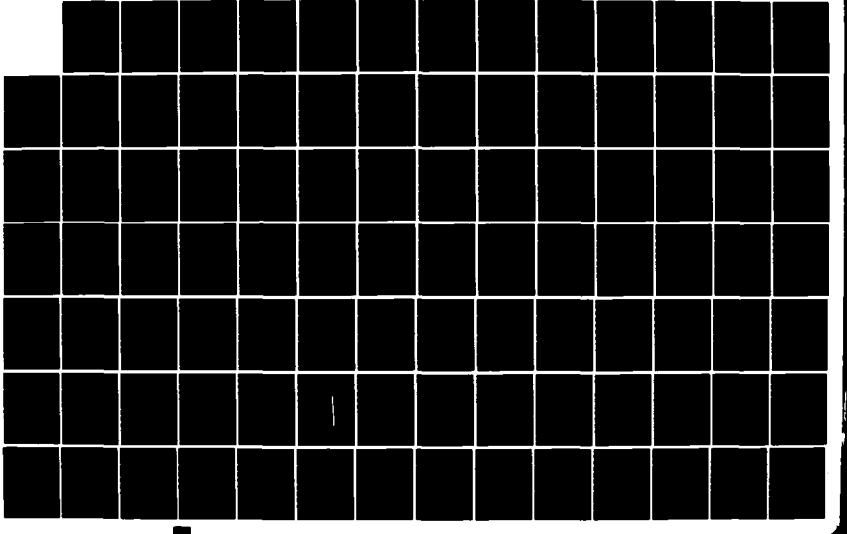
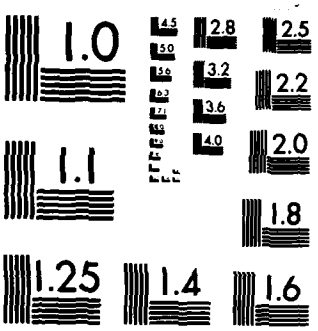
AD-A155 466 BAYESIAN VS MAP MULTIPLE MODEL ADAPTIVE ESTIMATION FOR FIELD OF VIEW EXPA..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. P A LOVING 1/4

UNCLASSIFIED MAR 85 AFIT/GE/EE/85M-1 F/G 12/1 NL





BAYESIAN VS MAP MULTIPLE MODEL

ADAPTIVE ESTIMATION FOR FIELD OF VIEW EXPANSION

IN TRACKING AIRBORNE TARGETS

THESIS

Phyllis A. Loving
Captain, USAF

AFIT/GE/EE/85M-1

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/EE/85M-1

BAYESIAN VS MAP MULTIPLE MODEL ADAPTIVE ESTIMATION FOR FIELD OF VIEW EXPANSION IN TRACKING AIRBORNE TARGETS

THESIS

Phyllis A. Loving
Captain, USAF

AFIT/GE/EE/85M-1

Approved for public release; distribution unlimited

AFIT/GE/EE/85M-1

BAYESIAN VS MAP MULTIPLE MODEL ADAPTIVE ESTIMATION FOR FIELD OF VIEW EXPANSION IN TRACKING AIRBORNE TARGETS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Phyllis A. Loving, B.S.

Captain, USAF

March 1985

Preface

This study is another link in a growing chain of research at the Air Force Institute of Technology to design a tracking algorithm for use with the Air Force Weapons Laboratory's high energy laser weapon system. As such, my work extends the multiple model adaptive filter developed by Lt. Robert I. Suizu and investigates Bayesian vs maximum a posteriori estimation using the multiple model adaptive filter concept.

I owe alot to my predecessors in this effort; without whom an investigation of this complexity could not have been attempted. My sincere gratitude goes to Dr. Peter S. Maybeck, my thesis advisor, for the effort and time expended on my behalf. His ability to consistently fan the flames of my enthusiasm and challenge my mind contributed significantly to keeping my work on track and productive. Finally, I thank my friends, classmates, and instructors whose sense of humor was there when mine was not.

Table of Contents

List of Figures

List of Tables

List of Symbols

Symbol	
AR	aspect ratio
$\underline{A}$	direction cosine matrix
A_p	area of a single pixel
A,B	atmosphere break frequencies
$\underline{a}$	vector of uncertain parameter values
α	smoothing constant in averaging process
$\alpha(t)$	azimuth
$\beta(t)$	elevation
$\underline{B}$	control input matrix
E	sample mean error
E{ }	expected value
$\underline{e}$	unit vector
e	error between truth model system output and filter system output
$\underline{F}$	system plant matrix
$F( )$	Fourier transform operation
$F^{-1}( )$	inverse Fourier transform operation
f_x, f_y	spatial frequencies
$\underline{G}$	system noise input matrix
G,H	complex-valued function in spatial frequency domain
$\bar{g}$,h	complex-valued function in the space domain
$\underline{h}[\ ]$	nonlinear intensity function
$\underline{H}[\ ]$	linearized intensity function
$\underline{H}$	system ouput matrix
$\underline{I}$	identity matrix

List of Symbols (Continued)

Symbol	
I_{max}	maximum intensity of the hot-spot
$\underline{K}$	Kalman filter gain
$\underline{\Phi}$	system state transition matrix
N x N	size of data array to be processed
ρ	range to target
$\underline{P}$	state covariance matrix or dispersion matrix of target image
p	conditional probability
$\underline{Q}_{FD}$	dynamics driving noise matrix
$\underline{Q}_{AD}$	noise covariance kernel descriptor for truth model dynamics
$\underline{R}$	measurement noise covariance matrix
$\underline{r}$	residual vector
r_h	horizontal range
σ	standard deviation of a process or dispersion of an intensity distribution
τ_{AF}	correlation time assumed for atmospheric jitter
τ_{DF}	correlation time assumed for target dynamics
t	time
Δt	sample time
θ	orientation angle in the FLIR image plane
$\underline{u}$	deterministic control input
$\underline{v}$	measurement noise vector - zero-mean, white Gaussian noise
$\underline{v}_I$	inertial velocity vector

Symbol	
v'	unit variance, zero mean, white, Gaussian noise vector
$\underline{w}$	white Gaussian noise vector
$\underline{x}$	general state vector
$\underline{x}_I, \underline{y}_I, \underline{z}_I$	inertial axes
ξ	angle between inertial velocity vector and the plane that is perpendicular to the line of sight
(x,y)	FLIR frame coordinates
$\hat{\underline{y}}(t)$	current averaged data frame
$\underline{y}(t)$	current data frame
$\underline{Z}$	measurement history
$\underline{z}$	measurement vector
	perpendicular
LOS	line of sight
rms	root mean squared
^{c}	Cholesky square root

Subscripts

A,a	atmospheric disturbances
AF	filter model for atmospheric jitter
α	azimuth direction
β	elevation direction
C	centroid
D	target dynamics
DF	filter model for target dynamics

Subscripts

d	discrete time
F,f	filter
I	inertial reference frame
i	i-th time frame
k	k-th term or filter
kl	kl-th pixel
m	m-th hot-spot or ellipsoid
MM	multiple model
n	normalized
o	initial value
pv	direction perpendicular to the target velocity
ppv	direction mutually perpendicular to the target velocity and pv direction above
r	direction along line of sight to the target from the tracker
T	truth model
v	direction of the target velocity vector

Superscripts

c	value after control application
^	estimate
-	before measurement update
+	after measurement update

Abstract

Previous efforts at the Air Force Institute of Technology have led to the development of a multiple model adaptive filter (MMAF) tracking algorithm which demonstrates significant improvements in performance against close-range, highly dynamic, airborne targets, over a direct correlation method currently in use. The basic elemental filter in the MMAF bank combines an enhanced correlator and a linear Kalman filter. Digital signal processing techniques are used to derive a target shape function from the forward looking infrared sensor data. This shape function is used as a template in the correlation algorithm which generates offset pseudo-measurements for the update portion of a linear Kalman filter. The multiple models are created by tuning the basic model for "best" performance against differing target maneuvering behavior and with physically different fields of view. The outputs of three independent elemental filters, each receiving data from a shared sensor are used to generate a single adaptive estimate of the state via a probabilistic weighted average (Bayesian form) or by selection of the one elemental filter associated with the highest probability (MAP form). The adaptive state estimate can produce target position predictions to be used in generating feedback control for maintaining the target in the center of the field of view.

There are two main results from this effort. The addition of a third elemental filter to the baseline MMAF improves tracking performance over the two-element MMAF. Specifically, the peak error following a maneuver is significantly reduced. However, the MAP estimation approach does not differ significantly from the Bayesian approach.

BAYESIAN VS MAP MULTIPLE MODEL ADAPTIVE ESTIMATION FOR FIELD OF VIEW EXPANSION IN TRACKING AIRBORNE TARGETS

I. INTRODUCTION

Since the inception of the Strategic Defense Initiative (SDI), increasing interest has been focused on laser weapon systems. The extremely tight specifications required for the pointing and tracking systems supporting these laser weapons has motivated research into innovative methods of accurately tracking targets maneuvering at high velocities.

1.1 *Background*

The Air Force Weapons Laboratory at Kirtland AFB, New Mexico, maintains ongoing research efforts using high energy laser weapons against airborne targets. Currently, pointing and tracking functions are accomplished by a correlation algorithm. A Forward Looking Infra-Red (FLIR) sensor makes target measurements, which the operational correlator uses, comparing data from the previous sample period to data available during the current sample time. Cross correlations of the data sets generate relative positions offsets, presumably due to target motion. These offsets are used to center the target in the FLIR field of view (FOV). Allowing for boresight error corrections, the laser is coupled to the FLIR so that positioning the center of the sensor FOV also points the laser toward the target.

Although the correlation tracker performs reasonably well against a wide variety of targets, it has several inherent limitations. The algorithm processes the most recent data, providing information on the current position of the target relative to the target position at the last sample time. Controls are applied some finite calculation time later. This delay causes pointing errors, which become increasingly detrimental with highly maneuverable targets. Additionally, the relative offsets are a function of image changes between one frame and the next, deviations not entirely due to target motion. These deviations can also be attributed to mirror vibration, atmospheric distortion, and inherent FLIR measurement errors. The correlator does not differentiate between the effects of these error sources and actual target motion. Research into alternate tracking algorithms is motivated by the above limitations. Since 1978, the Air Force Institute of Technology has supported a number of Master's theses demonstrating the feasibility and performance of a tracking algorithm based on Kalman filtering techniques.

The Kalman filter directly addresses the previously mentioned limitations, by attempting to separate actual target motion from other phenomena that are observed as apparent target motion. Using the statistical characteristics of atmospheric jitter and measurement errors, and a model of the anticipated target dynamics, the filter predicts an estimated target position. This prediction allows the FLIR to anticipate target motion, thus reducing tracking error due to time delays.

The initial feasibility study by Mercier [6], based on a four-state extended Kalman filter, assumed the distant target's intensity pattern on the FLIR image plane could be well modeled as a bivariate Gaussian distribution. This filter was based on a benign target dynamics model with target position represented by a zero mean, first order Gauss-Markov process. The filter's model for noise in the individual FLIR pixel intensity measurements was uncorrelated in time and space. This algorithm produced an order of magnitude improvement in tracking error over the correlation algorithm, in benign scenarios with filter parameters well-matched to the real world environment.

In robustness studies conducted by Harnly and Jensen [3], the spread, size, and shape of the target intensity pattern were varied to study the filter's sensitivities to incorrect assumptions. More dynamic target motion and adaptive estimation of target shape, still assuming bivariate Gaussian profiles, but with generally noncircular constant intensity contours, were incorporated and the filter's dynamic model was expanded to estimate velocity and acceleration. Spatial and temporal noise correlations were investigated, leading to the addition of a nearest-two-pixels spatial correlation for the filter's noise model. In an effort to track the maneuvering target, gain changes and other adaptations based on ad hoc responses to a maneuver detection were added to the filter. While the results were encouraging, the filter still experienced difficulty with harsh maneuvers differing from the assumed target dynamics model.

Up until this point, the filters have all assumed a known target intensity shape function with a single peak. In contrast, Singletery [12] and Rogers [11] developed signal processing techniques to derive the target shape function based on the intensity measurements. This technique was tested against multiple-hot-spot targets with dynamic variations. These efforts, concentrating on the shape function derivation, only pursued the benign target dynamics (as seen in Mercier's formulation). Rogers developed an alternative filter algorithm using the target shape function as a template for an enchanced correlator, rather than using it as a measurement function for an extended Kalman filter. The correlator produced offsets, x and y, from the center of the sensor FOV which were used as "measurement" inputs to a linear Kalman filter. The linear Kalman filter requires a much lower level of computational resources than the extended Kalman filter and would be preferable if comparable performance were achieved. In fact, comparable RMS tracking errors were produced in performance analyses, with the extended Kalman filter yielding a larger mean errors, while the linear filter/correlator combination produced larger standard deviations.

Follow-on research by Millner [8] and Kozemchak [4], incorporated more accurate target dynamics models. Both the extended Kalman filter and the linear filter/correlator algorithm were tested against close range targets, with maneuvers of up to 20 g's. Again, the extended Kalman filter exhibited larger biases and smaller standard deviations than the linear filter/correlator. However, both filter formulations

were slow to respond to harsh maneuvers that differed significantly from the filter's assumed target dynamics models and the tradeoff between precision tracking of benign targets versus maintaining lock on a highly maneuvering target was not completely resolved.

In an effort to address this limitation, Flynn [2] initially investigated a multiple model adaptive filter (MMAF). The attractiveness of this approach was increased by the potential for distributed processing. The MMAF was successfully implemented by Suizu [13]. Based on probabilistic weighting, the filter adaptively changed target dynamics model and FOV size, allowing the filter to maintain lock and track targets performing 20 g pull-ups at ranges of 20 km. This approach was tested using both the extended Kalman filter and the linear filter/correlator with both exponentially time-correlated acceleration and constant turn-rate dynamics models. The two types of filters continued the same relative performance as seen in [8] and [4], but were capable of tracking targets with trajectories that differed significantly in maneuver magnitude during the course of the tracking scenario.

1.2 Problem

This effort concentrates on expanding the multiple model linear/correlator algorithm developed by Suizu [13]. The potential for decreased computational loading compared to the extended Kalman filter, while maintaining comparable accuracy, makes this filter more attractive for further development. The model, as developed, will be operated at a 100 Hz sampling rate, as well as 30 Hz, to investigate the potential benefit of optical implementation of the algorithms discussed by Rogers

[11] and Roemer [10]. Additionally, the simulation will be extended beyond the current five seconds; both filters currently exhibit some difficulty with the minimum range/maximum crossing rate conditions at the 5 second point in the simulation, and this will allow exploration of tracking performance beyond this difficult period. To improve overall performance against highly maneuvering targets, another filter is incorporated in the MMAF tracker. Additionally, Maximum a Posteriori (MAP) estimation is compared to Bayesian estimation, i.e., producing an adaptive state estimate from the one model associated with the highest probability of validity rather than from a probabilistically weighted average of all models. This comparison explores the tradeoff between faster response to maneuvers and decreased accuracy of the estimate. Within the MMAF is a bank of N separate filters. Each of these can be an extended Kalman filter or a linear filter/correlator combination in this problem.

1.2.1 The Extended Kalman Filter Tracker. The extended Kalman filter algorithm uses measurements from the FLIR sensor to update the state estimate, which is propagated forward in time, based on the dynamics model of the Kalman filter and used to control the FLIR/laser orientation as shown in Figure 1-1. The measurement vector $\underline{z}(t_i)$ consists of the intensity output of 64 pixels arranged in an 8-by-8 "tracking window". The extended Kalman filter uses the linearized and nonlinear intensity functions ($\underline{H}[\hat{\underline{x}}(t_i^-),t_i]$ and $\underline{h}[\hat{\underline{x}}(t_i^-),t_i]$), along with the measurement vector $\underline{z}(t_i)$, to update the state estimate based on the equation

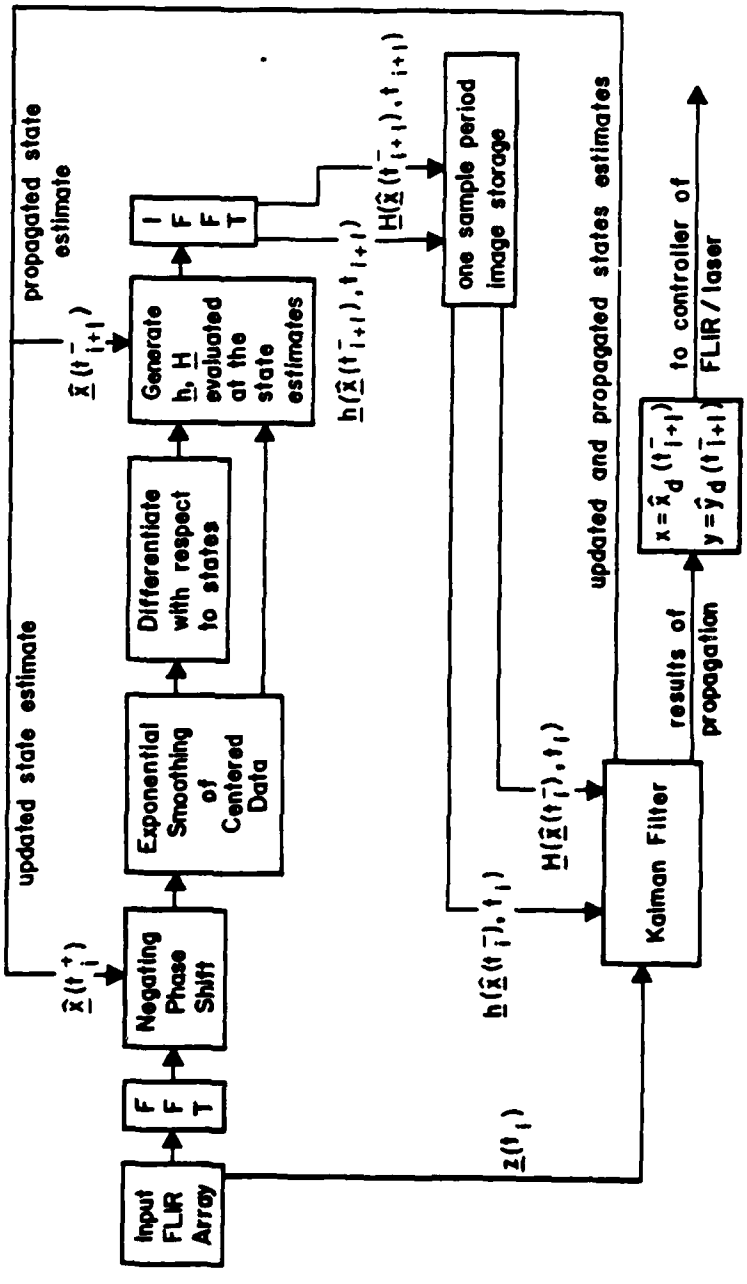


Figure 1-1. Data Processing Algorithm

$$\underline{\hat{x}}(t_i^+) = \underline{\hat{x}}(t_i^-) + \underline{K}(t_i)[\underline{z}(t_i) - \underline{h}(\underline{\hat{x}}(t_i^-), t_i)] \qquad (1\text{-}1)$$

where

$\underline{\hat{x}}(t_i^+)$ = state estimate after update

$\underline{\hat{x}}(t_i^-)$ = state estimate prior to update

$\underline{K}(t_i)$ = Kalman filter gain

$\underline{z}(t_i)$ = measurement vector; the assumed model is:

$$\underline{z}(t_i) = \underline{h}[\underline{x}(t_i), t_i] + \underline{v}(t_i)$$

$\underline{h}[\underline{\hat{x}}(t_i^-), t_i]$ = intensity shape function for measurements at t_i as a function of the state estimate

$\underline{H}[\underline{\hat{x}}(t_i^-), t_i]$ = $\partial \underline{h}/\partial \underline{x}$; the linearized intensity function evaluated at $\underline{\hat{x}}(t_i^-)$ at time t_i^- used to generate $\underline{K}(t_i)$

$\underline{v}(t_i)$ = measurement noise

Based on the internal dynamics models, the Kalman filter will propagate the estimate forward one sample period. The controller uses this information to orient the FLIR so that the center of the FOV is pointing at the predicted target position. It is assumed that the controller is capable of repositioning in less than one sample period.

The upper path of Figure 1-1 details the generation of the intensity shape functions at each sample point. First, the input data from the FLIR is Fourier transformed; the transformation is motivated by the possibility of optical processing and the comparative ease of performing the necessary computations in the frequency domain. The original 8-by-8 tracking window is expanded into a 24-by-24 data array, since the larger array reduces errors due to edge effects, aliasing, and

leakage conditions encountered in transforming a finite sequence [12:18]. In this application, the larger 24-by-24 array is produced by padding the 8-by-8 tracking data with additional intensity data available from the FLIR.

This transformed 24-by-24 data array is altered so as to center the target intensity pattern and then temporally averaged with the previous frames of transformed and centered data, to generate the desired nonlinear intensity function. This insures that any variations in the data are primarily due to noise and not changes in the target position in the FOV. The "centering" in the original domain is actually performed by multiplying the transformed data by a negating phase shift, based on the shifting property of the Fourier transform [9]. The negating phase shift is the complex conjugate of the linear phase shift due to the (estimated) target image offset in the spatial domain. This offset is obtained from the updated state estimates of the extended Kalman filter.

After centering, the data is filtered by exponential smoothing. This exponential smoothing approximates a true time averaging without the corresponding need to maintain numerous frames of data in memory. Because the noise is expected to vary more from frame to frame than the target intensity pattern, its effect is reduced by averaging successive frames of data. The resulting image is used as the target intensity shape function, $\underline{h}[\hat{\underline{x}}(t_i^-),t_i]$. The linearized intensity function, $\underline{H}[\hat{\underline{x}}(t_i^-),t_i]$, is obtained by applying the derivative property of the Fourier transform, which requires only a simple multiplication.

These linear and nonlinear shape functions are used to update the state estimates at the next measurement. To insure the proper location of the shape function in the FOV, both functions are evaluated using the propagated state estimates, $\hat{\underline{x}}(t_{i+1}^{-})$. It is assumed that the FLIR is centered on the position predicted by the target dynamics model. Thus, the intensity functions are evaluated at the position predicted by the atmospheric states, if there is a single elemental filter; this must be modified in the multiple model adaptive filter, and the modification is discussed in Section 4.4. The Fourier transform shifting property is applied again to phase shift the transformed image by the estimated atmospheric offset. Finally, the inverse Fourier transform is performed, returning the intensity functions to the spatial domain for use in the next update cycle.

1.2.2 Linear Kalman Filter/Correlation Tracker. The linear filter/correlator developed by Rogers [11] is very similar to the extended Kalman filter algorithm depicted in Figure 1-1. The evaluation of $\underline{H}$ is no longer required; but otherwise, the upper path remains the same. The $\underline{h}$ and $\underline{z}$ are not used directly by the Kalman filter. The intensity function, $\underline{h}[\hat{\underline{x}}(t_{i+1}^{-}), t_{i+1}]$, is used as a template by an enhanced correlator algorithm. Sensor measurements are compared to the template, producing estimated offsets between the target centroid and the center of the sensor FOV. These two scalar offsets are pseudo-measurement inputs to a linear Kalman filter (the filter of Section 1.2.1 was nonlinear only in the measurement update portion of the algorithm when a linear state propagation model is used). In all other respects, the two tracking algorithms are the same.

1.3 Overview

The following chapters expand the details of the algorithm, the testing environment, and the results. Chapter II presents the mathematical foundation for the multiple model algorithm for Bayesian estimation and MAP estimation. The truth model is covered in Chapter III, establishing the standard for performance evaluation. Chapter IV describes the basic linear filter/correlator model in detail along with the MMAF implementation. Chapter V covers the tracker statistics, parameters, plot formats, and the test scenarios. Finally, Chapter VI contains the performance analysis. The conclusions and recommendations are in Chapter VII.

II. THEORETICAL DEVELOPMENT OF MULTIPLE MODEL ADAPTIVE ESTIMATION

From a Bayesian point of view, it would be desirable to be able to generate the conditional density function of the state given the measurement history; with this density function, a good estimate of the state vector, $\underline{x}$, is obtained by evaluating $E\{\underline{x}(t_o)|\, Z_i\}$, the conditional expectation of the state given the measurement history. If the target motion is adequately described by the linear Kalman filter dynamics model, this conditional mean is easily calculated. However, in this problem the appropriate target dynamics model may be uncertain and/or time varying. Let $\underline{a}$ be the vector of uncertain parameters in the Kalman filter dynamics model. In order to estimate the state vector properly, it is also necessary to estimate $\underline{a}$, thus motivating the evaluation of $f_{x,a|Z}$, the joint density of $\underline{x}$ and $\underline{a}$ given the measurement history. The evaluation of this density function is the core of Bayesian estimation [5:129].

2.1 Mathematical Development

Application of Bayes theorem to the joint conditional density function of the variables to be estimated, given the measurements, yields:

$$f_{x,a|Z} = f_{x|a,Z} \cdot f_{a|Z} \tag{2-1}$$

The first term on the right hand side of the equal sign is a Gaussian distribution, if the adequate system models are in the form of a linear

state differential and linear measurement equation, all driven by white Gaussian noises. As such, it is totally specified by its mean and covariance, which are the outputs of a Kalman filter based on one particular value of $\underline{a}$. The second term can be expressed as

$$f_{\underline{a}|Z_i} = \frac{f_{z_i|\underline{a},Z_{i-1}} \cdot f_{\underline{a}|Z_{i-1}}}{\int_{\infty}^{\infty} f_{z_i|\underline{a},Z_{i-1}} \cdot f_{\underline{a}|Z_{i-1}} \, d\underline{\alpha}} \tag{2-2}$$

where $\underline{\alpha}$ is a dummy variable associated with $\underline{a}$ values, and

$$\underline{Z}_i = \begin{bmatrix} \underline{Z}_{i-1} \\ \\ \underline{z}_i \end{bmatrix}$$

and

$\underline{Z}_{i-1}$ = the composite vector of all measurements prior to time t_i

z_i = the vector of measurements available at time t_i

The first numerator term in Equation (2-2) is a Gaussian density function with mean $\underline{H}(t_i)\hat{\underline{x}}(t_i^-)$ and covariance $[\underline{H}(t_i)\underline{P}(t_i^-)\underline{H}^T(t_i)+\underline{R}(t_i)]$, calculated by a Kalman filter based on a given value of $\underline{a}$. Starting with $f_{\underline{a}|Z} = f_{\underline{a}}(\alpha)$, the a priori probability of a particular dynamics model correctly modeling the actual target motion at the initial time (letting $\underline{a}$ determine the applicable dynamics model), Equation (2-2) can be calculated recursively. Having evaluated both terms in Equation (2-1), the desired conditional mean $E\{x(t_i)\}Z_i$, is readily determined.

$$E\{\underline{x}(t_i)|Z_i\} = \int_{-\infty}^{\infty} \xi f_{x|z}\, d\xi$$

$$= \int_{-\infty}^{\infty} \xi \left[\int_{-\infty}^{\infty} f_{x,a|z}\, d\alpha\right] d\xi$$

$$= \int_{-\infty}^{\infty} E\{\underline{x}|\underline{a},\underline{Z}_i\} \; f_{a|z}\, d\alpha$$

This result is based on Equation (2-1), the concept of marginal densities, Bayes' theorem, and changing the order of integration. While the above calculations achieve the desired estimate, they are computationally unattractive. On-line implementation of the integrals is difficult and time consuming, if at all tractable. The additional requirement for an infinite number of Kalman filters to span the continuous $\underline{a}$ parameter space suggests that a judicious discretization might provide a more feasible approach.

Discretization of the parameter space allows the integrals over $\underline{a}$ to be evaluated as the summation of all possible $\underline{a}$ values. Assume the parameter $\underline{a}$ can take on a finite number of values, selected to span the set of possible values in the continuous parameter space and represented by the set $\{\underline{a}_1, \underline{a}_2, \ldots \underline{a}_N\}$. Each element $\underline{a}_k$ represents a different system model, which is ultimately reflected, in this application, as a separate Kalman filter based on the dynamics model (and appropriate FOV size) defined by $\underline{a}_k$. Bayesian estimation, using the outputs of the N filters, requires an a priori density function for $\underline{a}$ as well as $\underline{x}(t_o)$. Let $p_k(t_o)$ represent the probability that the model specified by $\underline{a}_k$ is the best representation of the true target dynamics at the initial time. Also define the hypothesis conditional probability $p_k(t_i)$ as

$$p_k(t_i) = \text{pr}\{\underline{a} = \underline{a}_k | \underline{Z}_i = Z_i\} \qquad (2\text{-}4)$$

The development of the recursive calculation of this value and the conditional mean and covariance of the state, given the measurement history, follows directly from Equation (2-2) and Equation (2-3), noting that $f_{a|Z}$ is comprised of a sum of weighted Dirac delta functions weighted by $p_k(t_i)$

$$p_k(t_i) = \frac{f_{z_i|a_k,Z_{i-1}} \cdot p_k(t_{i-1})}{\sum_{j=1}^{N} f_{z_i|a_j,Z_{i-1}} \; p_j(t_{i-1})} \qquad (2\text{-}5)$$

$$\text{for } k = 1,2,\ldots,N$$

The recursion for $p_k(t_i)$ is expressed in terms of $p_k(t_{i-1})$ and other elements that can be evaluated. In terms of these $p_k(t_i)$ values, Equation (2-3) becomes

$$\begin{aligned} \hat{\underline{x}}(t_i^+) &= \sum_{k=1}^{N} E\{\underline{x}(t_i)|\underline{a}_k,Z_i\} \; p_k(t_i) \\ &= \sum_{k=1}^{N} \hat{\underline{x}}_k(t_i^+) \; p_k(t_i) \end{aligned} \qquad (2\text{-}6)$$

where $\hat{\underline{x}}_k(t_i^+)$ is produced by the Kalman filter that assumes the parameter vector equals $\underline{a}_k$. The denominator of Equation (2-5) is a normalizing constant that is the same for each conditional probability, $p_k(t_i)$, and is the sum of all the numerator terms. If these calculations are implemented by distributed processing, the numerators of Equation (2-5), for k = 1,2,...,N, can be calculated in parallel, along with each of the N separate filters. An estimate of $\underline{a}$ can be calculated but is not required to calculate state estimates. The conditional covariance of $\hat{\underline{x}}(t_i)$ is

$$\underline{P}(t_i^+) = \sum_{k=1}^{N} p_k(t_i) \{ \underline{P}_k(t_i^+) + [\hat{\underline{x}}_k(t_i^+) - \hat{\underline{x}}(t_i^+)] \cdot [\hat{\underline{x}}_k(t_i^+) - \underline{x}(t_i^+)]^T \} \tag{2-7}$$

where $\underline{P}_k(t_i^+)$ is the state error covariance calculated by the Kalman filter based on $\underline{a}_k$. If $\underline{P}_k(t_i^+)$ is desired, it must be computed on-line because it depends on the actual measurement history; however, it is not essential to the operation of the on-line algorithm. As seen in Equation (2-6), the state estimate resulting from this development is the probabilistically weighted average of the state estimates generated by N separate Kalman filters.

2.2 The Multiple Model Adaptive Filter

The multiple model adaptive filtering (MMAF) algorithm is based on the previous development. As depicted in Figure 2-1, the MMAF is composed of N separate Kalman filters, each based on a discrete value, $\underline{a}_k$, of the parameter vector. At time t_i, the measurement $\underline{z}_i$ is processed, generating residuals $r_1(t_i)$, $r_2(t_i)$,...$r_N(t_i)$. The residuals are used to evaluate the hypothesis conditional probabilities, $p_k(t_i)$, according to Equation (2-5). The probability $p_k(t_{i-1})$ needed for this iteration is maintained from the last sample period and

$$f_{z_i|a_k,Z_{i-1}} = \frac{1}{(2\pi)^{m/2} |\underline{A}_k(t_i)|^{1/2}} \exp\{*\} \tag{2-8}$$

$$\{*\} = \{-1/2\ \underline{r}_k^T(t_i) \underline{A}_k^{-1}(t_i) \underline{r}_k(t_i)\}$$

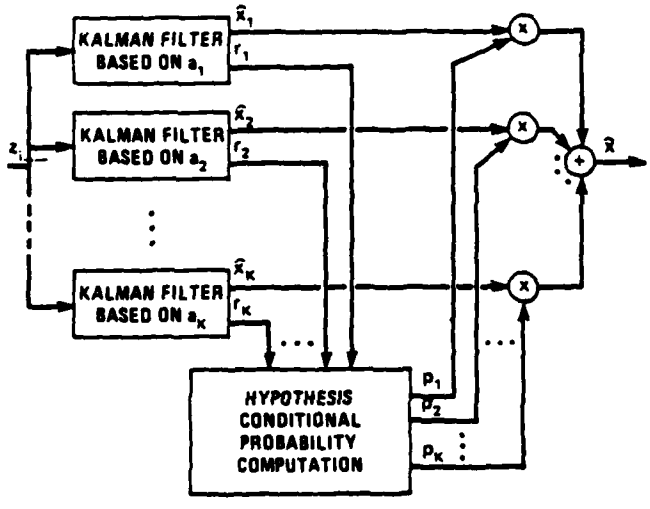


Figure 2-1. Multiple Model Filtering Algorithm

where m is the number of measurements and $\underline{A}_k(t_i)$ is available from the kth filter as

$$\underline{A}_k(t_i) = \underline{H}_k(t_i)\underline{P}_k(t_i^-)\underline{H}_k^T(t_i) + \underline{R}_k(t_i) \tag{2-9}$$

Finally, the adaptive state estimate is generated as the probablistically weighted average seen in Equation (2-6), portrayed as the output of the summing junction in Figure 2-1.

One expects that the filter that most nearly represents the true target motion will produce a true residual covariance very close to its internally computed $\underline{A}_k(t_i)$. The "mismatched" filters should produce larger residuals than anticipated, so that Equation (2-5) leads to a heavier weighting of the estimates produced by the "best" filter. The algorithm's performance depends on the existence of significant differences between the residuals of the "correct" filter and the "mismatched" filters. These significant differences can be enhanced by specifically tuning each filter for its best performance against a target trajectory that matches its internal dynamics model. The common practice of "conservative" tuning of a single nonadaptive Kalman filter by adding additional dynamics pseudo-noise should be avoided, since it tends to blur the distinctions between the estimates and residuals based on different models.

Additionally, the calculated probabilities should have an artificially enforced lower bound. By bounding the lower value, the p_k of a "bad" filter is prevented from converging to zero, which would effectively remove that filter from the bank and prevent the MMAF from

responding as accurately or rapidly to future changes in the parameter values. After bounding, the p_k values are rescaled so that the probabilities still sum to one. The selection of the lower bound value can significantly affect the accuracy of the estimate due to inappropriate weighting of the "mismatched" filters. A bound of 0.001 is selected for this effort.

In this application, the MMAF consists of a bank of either two or three elemental filters. Each elemental filter has the same form; however, they are tuned for different assumed target trajectories. Each filter acquires data from a wide tracking window (24-by-24 pixels) or a narrow tracking window (8-by-8 pixels), allowing the filter to maintain lock on a highly dynamic target or providing greater resolution for more benign target motion, respectively. Specifically, the two-filter MMAF's first filter is optimized for a benign trajectory and an associated small tracking window, and the second filter, for a 20 g pull-up target maneuver with a larger tracking window being used. The three-filter MMAF contains the first two elemental filters plus a narrow-FOV filter tuned to perform well against 10 g maneuvers. By changing the probabilistic weights, the MMAF adaptively changes tracking window size and dynamics model. The current two-filter MMAF has a very coarsely discretized parameter space; this results in poorer performance against trajectories which are not modeled by one of the elemental filters. A finer discretization, yielding three or more elemental filters, plus the added ability to restart a diverging element filter (using estimates from the nondiverging filter(s) to accomplish the "restart") should produce more accurate tracking than purposefully detuning the elemental filters to prevent divergence of the MMAF when tracking a 20 g maneuver.

2.3 Maximum A Posteriori Estimation Algorithm

Maximum a posteriori (MAP) state estimation is based on calculating the mode of $f_{x|Z}$. Under the standard linear system, Gaussian noise assumptions for a Kalman filter development, the conditional mean, $\hat{\underline{x}}(t_i^+)$, is the mode. However, due to the uncertain character of the parameter vector $\underline{a}$, one must consider the joint conditional density function, $f_{x,a|Z}$. In this case, it is reasonable to assume that the conditional mean (or mode), calculated by the filter based on the most probable $\underline{a}_k$ (the one with the highest p_k) is a good approximation to the true mode of $f_{x,a|Z}$. This is not just an approximation if $\underline{a}$ is truly discrete-valued.

The MAP multiple model filter calculates $p_k(t_i)$ as shown in Section 2.2. However, unlike the Bayesian MMAF, the MAP estimate is not the optimally weighted average of the elemental filter estimates. Instead, it is assumed that the single filter with highest $p_k(t_i)$ is the most likely representation of the true target behavior. The estimate, $\hat{\underline{x}}_k(t_i^+)$, produced by this single filter is used as the MAP MMAF estimate until another filter becomes the "best" match to the actual target dynamics.

The MAP approach to estimation has positive and negative aspects as compared to the Bayesian approach. It is expected that the MAP estimator will respond more rapidly to changes in target maneuver characteristics. Moreover, when the true target motion is well matched to one of the filters in the bank, estimation accuracy is not degraded significantly by including information from the "mismatched" filters. However, since the need to reduce computational loading requires a coarse

discretization of the parameter space, the true target behavior often is not well matched to any one filter in the bank. In this case, the estimator operates with an inappropriate dynamics model and can be expected to provide degraded performance. The Bayesian estimator provides higher accuracy when none of the models match the true target behavior because of its probabilistically weighted averaging. Each of the approaches are based on sound reasoning, leaving experimental results to determine which approach provides the best results for a given problem.

III. TRUTH MODEL

3.1 Introduction

The truth model provides the simulation's standard of performance; the standard to which the filter estimates are compared and the "reality" from which the measurement input is constructed. As such, it should provide a close approximation to the real world environment, particularly in the areas critical to the problem of concern. The processes of importance in this investigation are atmospheric jitter, target dynamics and shape effects, and background and FLIR noises. Vibrational effects can also be significant, but are neglected here, based on the assumption of a ground based weapon system. These processes are important because they contribute to apparent target motion as observed by the sensor.

Apparent target motion is described using an x-y coordinate system in the FLIR image plane, expressed in units of pixels (a pixel is 20 μrads-by-20 μrads). The coordinates of the observed target centroid are the sum of the offsets from the center of the FOV due to actual target motion and the offsets due to atmospheric disturbances

$$x_C = x_D + x_A$$

where

x_C - observed centroid x coordinate

x_D - x coordinate of the offset due to dynamics

x_A - x coordinate of the offset due to atmospherics

and similarly for y.

Target motion in the truth model is described by a linear stochastic differential equation [5:163]

$$\dot{\underline{x}}(t) = \underline{F}\,\underline{x}(t) + \underline{B}\,\underline{u}(t) + \underline{G}\,\underline{w}(t)$$

where the state vector, $\underline{x}(t)$, is composed of the target position and atmospheric states. The $\underline{Bu}$ term consists of deterministic velocities applied to move the target in a preselected inertial trajectory. The white Gaussian noise term drives the atmospheric jitter "shaping filter" to generate x_A and y_A.

The measurements used by the filter algorithm are derived from the truth states using the target image intensity function developed later in this chapter. Values derived from this intensity function are corrupted by adding spatially correlated and temporarily uncorrelated noise, accounting for FLIR and background noises. The 64-element vector thus constructed is compared to a template, derived from earlier data, by the correlator portion of the tracking algorithm to produce the offset values used as measurement input to the linear Kalman filter. The measurement is related to the state variables by the following discrete-time linear stochastic equation

$$\underline{z}(t_i) = \underline{H}\,\underline{x}(t_i) + \underline{v}(t_i)$$

The target model and the measurement model incorporated in the same simulation truth model are described in this chapter, which covers target dynamics, atmospheric effects, the various coordinate frames,

the transformation of inertial motion to the FLIR image plane coordinates, and the inertial target trajectories used for this effort. The target intensity function, the target image projection, and noise effects are defined in greater detail as well. Significant aspects of the truth model could be replaced with actual data; however, this approach of simulating a truth model provides control over many aspects not typically available from test data.

3.2 Truth State Model

The truth model describes the apparent motion due to dynamics and atmospheric effects. The target dynamics model, developed by Harnly and Jensen [3], provides the true location of the target center of mass in the FLIR image plane horizontal and vertical directions. Additionally, this section presents the equations which translate a simulated target inertial trajectory into two dimensional motion on the FLIR image plane.

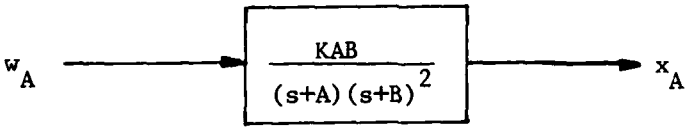
In order to evaluate specific trajectories, a deterministic model provides the target location time history. The first two elements of the state vector, x_1 and x_2, are the dynamic states in the horizontal and vertical directions in the FLIR image plane. Because of the large distances, the azimuth angle, α , and elevation angle, β , are essentially the linear coordinates, x and y, of the target centroid position in this FLIR image plane. More details regarding this relation are provided later in this chapter. In order to conform to the linear differential equation format, the deterministic inputs must be $\dot{\alpha}$ and $\dot{\beta}$ instead of α and β directly. Based on

$$\dot{x}_1(t) = \dot{\alpha}(t)$$
$$\dot{x}_2(t) = \dot{\beta}(t)$$

the model's dynamic motion can be expressed as

$$\dot{\underline{x}}_D(t) = \underline{u}(t) = [\dot{\alpha}(t) \quad \dot{\beta}(t)]^T \tag{3-1}$$

The atmospheric disturbances, developed by Mercier [6], are modeled as third order Gauss-Markov processes in both the horizontal (x) and vertical (y) image plane directions. In the x-direction this yields



where

w_A = unit strength, zero-mean, white Gaussian noise

K = system gain, adjust for desired atmospheric RMS value

A = break frequency; 14.14 rads/sec

B = break frequency; 659.5 rads/sec

x_A = output of the shaping filter

Thus, the atmospheric jitter is represented in x and y directions by the stochastic differential equation

$$\dot{\underline{x}}_A(t) = \underline{F}_A \, \underline{x}_A(t) + \underline{G}_A \, \underline{w}_A(t) \tag{3-2}$$

where

$\underline{x}_A(t)$ = six atmospheric noise states

$\underline{F}_A$ = atmospheric plant matrix

$\underline{G}_A$ = atmospheric noise input matrix

$\underline{w}_A(t)$ = vector of white Gaussian noise inputs with statistics

$$E\{\underline{w}_A(t)\} = \underline{0}$$

$$E\{\underline{w}_A(t)\underline{w}_A^T(t+\tau)\} = \underline{Q}_A(t)\ \delta(\tau)$$

The atmospheric states are augmented with the dynamic states, producing an augmented state differential equation with components given by Equations (3-1) and (3-2). The augmented system is then converted to the equivalent discrete time form [3]. The propagation of the target motion takes the form

$$\underline{x}(t_{i+1}) = \Phi(t_{i+1},t_i) + \begin{bmatrix} \underline{B}_d(t_i) \\ \cdots\cdots \\ \underline{0} \end{bmatrix} \underline{U}_d(t_i) + \begin{bmatrix} \underline{0} \\ Q_{Ad} \end{bmatrix} \underline{w}_{Ad}(t_i) \qquad (3\text{-}3)$$

where

$\underline{x}(t_i)$ = state vector (two dynamic states and six atmospheric states)

$\underline{B}_d(t_i)$ = 2-by-2 dynamics input matrix = I Δt, where $\Delta t = t_{i+1} - t_i$

$\underline{U}_d(t_i)$ = piecewise constant function (between sample times) evaluated at midpoint to approximate the integral of $\alpha(t)$ and $\dot{\beta}(t)$ from t_i to t_{i+1}

$$\underline{U}_d(t_i) = \begin{bmatrix} \dot{\alpha}(t_i + \Delta t/2) \\ \dot{\beta}(t_i + \Delta t/2) \end{bmatrix}$$

$\underline{w}_{Ad}(t_i)$ = discrete-time white Gaussian noise

$$E\{\underline{w}_{Ad}(t_i)\} = \underline{0}$$

$$E\{\underline{w}_{Ad}(t_i)\underline{w}^T_{Ad}(t_j)\} = \underline{I}\,\delta_{ij}$$

and

$$\sqrt[c]{\underline{Q}_{Ad}(t_i)} \qquad \sqrt[c]{\underline{Q}_{Ad}(t_i)}^T = \underline{Q}_{Ad}(t_i) \qquad [2]$$

where $\sqrt[c]{\underline{Q}_{Ad}}$ represents the Cholesky square root of $\underline{Q}_{Ad}$. [5] .

The state transition matrix is

$$\Phi(t_{i+1},t_i) \quad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{-A\Delta t} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{-B\Delta t} & te^{-B\Delta t} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{-B\Delta t} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-A\Delta t} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-B\Delta t} & \Delta te^{-B\Delta t} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-B\Delta t} \end{bmatrix}$$

Equation (3-3) is developed in detail in reference [3].

As stated earlier, azimuth velocity, $\dot{\alpha}(t)$, and elevation velocity, $\dot{\beta}(t)$, are derived from deterministic inertial velocities. The inertial velocities must be projected into the FLIR image plane coordinates, based on the geometry in Figure 3-1.

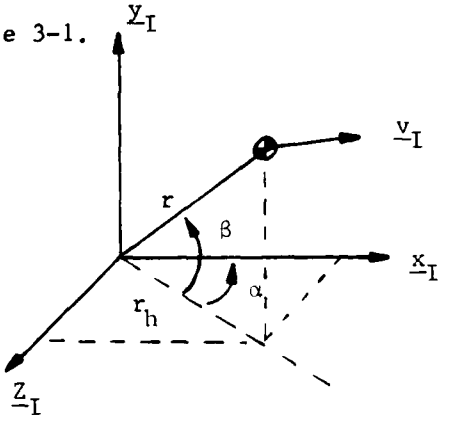


Figure 3-1. Inertial Coordinate Frame

where

$\underline{x}_I$, $\underline{y}_I$, $\underline{z}_I$ = inertial axes

r = range from tracker origin to target

r_h = horizontal range

$\underline{v}_I$ = target inertial velocity

α = azimuth angular displacement

β = elevation angular displacement

The geometry associated with azimuth direction is shown in Figure 3-2.

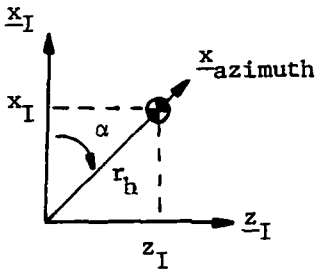


Figure 3-2. Azimuth Geometry

from Figure 3-2

$$\alpha(t) = \tan^{-1}\left[\frac{z_I(t)}{x_I(t)}\right] \quad \text{(rads)} \tag{3-4}$$

and so

$$\dot{\alpha}(t) = \frac{x_I(t)\dot{z}_I - z_I(t)\dot{x}_I(t)}{z_I^2(t) + x_I^2(t)} \quad \text{(rads/sec)} \tag{3-5}$$

The azimuth velocity from Equation (3-5) is in rads/sec, which must be converted to pixels/sec by dividing by 20×10^{-6} rads/pixel [3:33].

Similarly, Figure 3-3 illustrates the geometry involved in computing the elevation velocity.

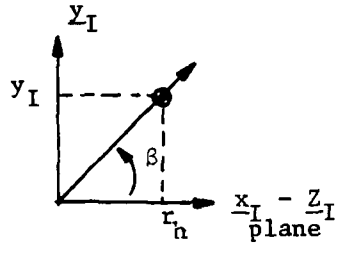


Figure 3-3. Elevation Geometry

where

$$r(t) = \text{range} = [x_I^2(t) + y_I^2(t) + z_I^2(t)\ ]^{1/2}$$

$$r_h(t) = \text{horizontal range} = [x_I^2(t) + z_I^2(t)\ ]^{1/2}$$

and

$$\beta(t) = \tan^{-1}\left[\frac{y_I(t)}{r_h(t)}\right] \quad \text{(rads)} \tag{3-6}$$

so that

$$\dot{\beta}(t) = \frac{r_h(t)\dot{y}_I(t) - y_I(t)\dot{r}_h(t)}{r^2(t)} \quad \text{(rads/sec)} \tag{3-7}$$

where, from the $r_h(t)$ expression above, it can be seen that

$$\dot{r}_h(t) = \frac{x_I(t)\dot{x}_I(t) + z_I(t)\dot{z}_I(t)}{r_h(t)}$$

Once again, the velocity must be converted to pixels/sec.

By substituting Equations (3-5) and (3-7) into Equation (3-1), the truth model generates target motion for the desired trajectory.

3.3 Target Trajectories

A number of deterministic trajectories are available which incorporate several different maneuver options. These trajectories are designed to provide realistic target behavior with fairly simple models. The basic equations are described in detail by Millner [8].

Trajectory one - This trajectory, as shown in Figure 3-4, is very benign. The target flies a constant-heading, straight-and-level course. The inertial velocity, $\underline{v}_I$, is constant for the entire maneuver and is parallel to the x_I-z_I plane.

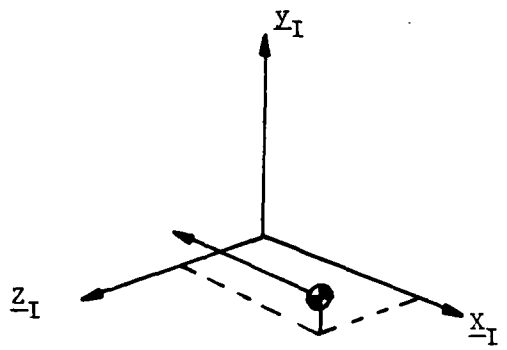


Figure 3-4. Trajectory One

Either wings-level flight or a constant roll-rate maneuver (positive roll rate defined by the right hand rule) may be simulated.

Trajectory Two - In order to evaluate the filter response to more dynamic behavior, this trajectory simulates a constant-g pull-up. The target initially flies trajectory one (wings level, one-g flight) until $t = 2.0$ seconds, allowing the filter to obtain good position estimates before the maneuver is initiated.

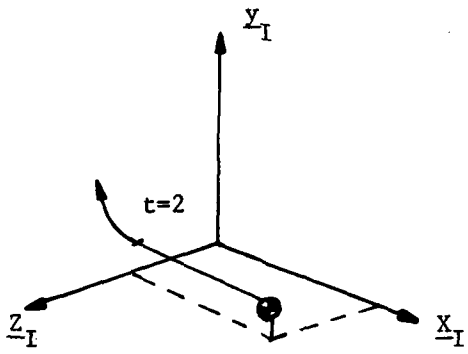


Figure 3-5. Trajectory Two

The pull-up is started with a step change to the pitch rate. While this is an unrealistic behavior, it is more harsh than maneuvers encountered in the real world, and the tracker should perform at least as well against real world targets.

Trajectory Three - This trajectory contains two maneuver changes, providing a means to evaluate tracking performance against a target that begins and ends a pull-up maneuver. As with trajectory two, a constant-g pull-up is executed; however, the pull-up is terminated at t = 3.5 seconds, prior to the end of the simulation. The target acceleration is impulsively set to zero; the inertial velocity the target has at that point remains constant until the end of the simulation.

Trajectory Four - This trajectory displays motion in all three inertial directions. Like trajectory two, the target initially flies straight and level and then, at t = 2.0 seconds, the target performs a constant-g pull-up, but in the $-z_I$ direction rather than the $+y_I$ direction (see Figure 3-5). The projected target image on the FLIR image plane changes more dramatically in this out-of-plane maneuver. This variation allows investigation of the filter's response to dynamic changes in the orientation and spacing of the individual hot-spots that compose the target intensity profile on the FLIR image plane.

3.4 Coordinate Frames

The dynamics of various parts of the target relative to its center of mass must be known to project the target image accurately onto the FLIR plane. This projection is required to calculate the intensity values for measurement inputs. Definition of two additional coordinate frames facilitates these calculations.

Target Frame - The origin of the target frame is the target's center of mass. The first axis is coincident with the velocity vector. Perpendicular to the first, the second axis points out the right side of the target. The final axis completes the right hand system and points out the underside of the target fuselage. In this simulation, the multiple hot spots are assumed to lie in the $\underline{e}_v$-$\underline{e}_{pv}$ plane. The target coordinate system is expressed as unit vectors $\underline{e}_v$, $\underline{e}_{pv}$, and $\underline{e}_{ppv}$ (v: direction of the velocity vector, pv: perpendicular to velocity, ppv: perpendicular to both).

$\alpha - \beta$ plane - The origin of this frame is also the target center of mass. One basis vector, $\underline{e}_r$, is aligned with the true line of sight from the tracker (located at the inertial coordinate system origin). The plane of interest is defined by unit vectors $\underline{e}_\alpha$ and $\underline{e}_\beta$ which are rotated from the inertial frame by the angles α and β, as shown in Figure 3-1.

3.5 Measurement Model

The measurements developed for the tracking filter are derived from the projection of the target intensity function onto the FLIR image plane. FLIR and background noises corrupt the intensity function. At close range, targets can be well modeled as the sum of bivariate Gaussian functions with elliptical contours [3]. As shown in Figure 3-6, the apparent target image eventually resolves into separate hot spots.

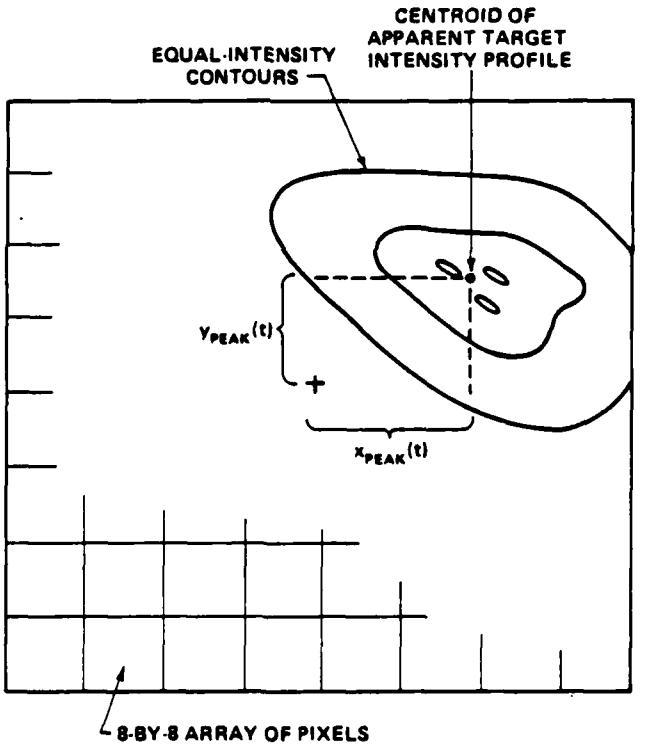


Figure 3-6. Apparent Target Intensity Pattern on FLIR Image Plane

This simulation used three identically distributed hot spots, each of which is described by the following intensity function.

$$I\,[x,y,x_{peak}(t),y_{peak}(t)] = I_{max}\,\exp\{-0.5\,[x-x_{peak})\;(y-y_{peak})] \cdot [\underline{P}]^{-1}\,[x-x_{peak})\;(y-y_{peak})]^{T}\} \tag{3-8}$$

where

I_{max} - maximum intensity of the hot spot

x_{peak}, y_{peak} - coordinates of the peak intensity of the hot spot

$\underline{P}$ - matrix whose eigenvalues are σ_v^2 and σ_{pv}^2, which are the dispersions of the elliptical constant-intensity contours in the target frame, and whose eigenvectors define the orientation of the ellipse principal axes

The x- and y-coordinates in this function are calculated in pixels relative to the center of the tracker field of view.

The location of the hot spots are expressed in the target coordinate frame. The intensity function centroid for single hot spot targets are assumed coincident with the target center of mass. In this study, the multiple hot spots are distributed as shown in Figure 3-7 [8:40].

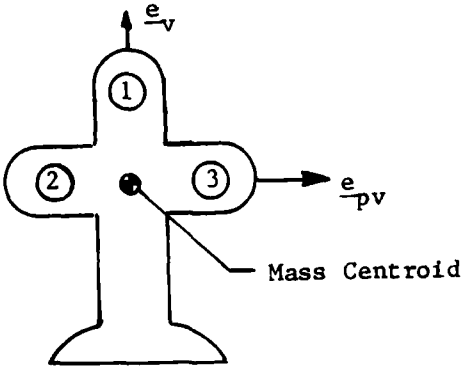


Figure 3-7. Hot Spot Distribution

Hot spot one is located at 1.0 meters in the $+\underline{e}_v$ direction; two and three are located at ± 0.5 meters in the $\underline{e}_{pv}$ direction. It is assumed that the semi-major axes of the ellipses are parallel and aligned with the velocity vector. The velocity vector remains out the nose of the target for one entire simulation. While these are not realistic assumptions, they facilitate the simulation of target dynamics, and the tracking algorithm can be expected to perform as well against live targets.

The measurements for both single and multiple hot spot cases are the average intensity values from each pixel of an 8-by-8 tracking window. This intensity is the sum of each hot spot contribution and background and FLIR noises. The measurement value for pixel kl:

$$z_{kl}(t_i) = \sum_{m=1}^{M} \{1/A_p \int_{\text{pixel kl}} I_m[x,y,x_{peakm}(t_i),y_{peakm}(t_i)]dxdy\} + v_{kl}(t_i) \qquad (3\text{-}9)$$

where

$I_m[*]$ = intensity function of the m^{th} hot spot of M total hot spots

$z_{kl}(t_i)$ = output of the kl^{th} pixel (k^{th} row, l^{th} column) at time t_i; the average intensity at that pixel as sensed by a detector in the FLIR image plane.

A_p = area of one pixel

(x,y) = coordinates of any point within the kl^{th} pixel

(x_{peakm}, y_{peakm}) = location of one peak of the m^{th} intensity function at t_i

$v_{kl}(t_i)$ = additive FLIR and background noises for the kl^{th} pixel

3.6 Target Image

During the course of the simulation, the target size and hot spot distribution on the target will not change; however, as the target maneuvers and approaches the sensor, the target image on the FLIR sensor does vary. In order to simulate these variations, the target image at any given time is referenced to a previously defined standard image. The target image is developed in the target frame and then projected onto the $\alpha - \beta$ plane, described in Section 3.3; the geometry is illustrated in Figure 3-8.

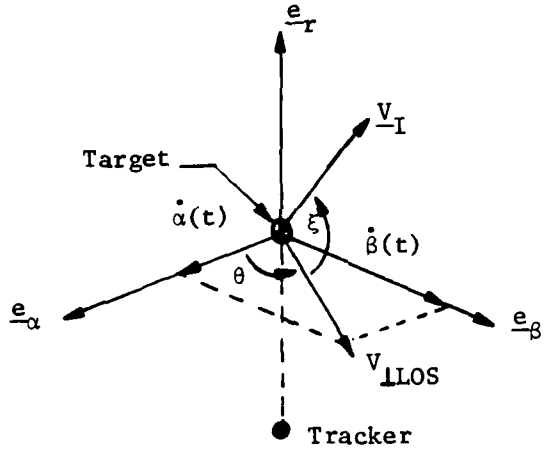


Figure 3-8. Image Projection [9:II-23]

The reference image is defined with the target flat in a plane perpendicular to the tracker line of sight. Any other orientation would produce a smaller image. The following expressions relate the current image size to the reference image based on the current range and velocity of the target [9:II-24].

$$\sigma_{pv} = \sigma_{pvo} \ (\rho_o/\rho)$$

$$\sigma_v = (\rho_o/\rho) \ \left[\sigma_{pvo} + (\sigma_{vo} - \sigma_{pv}) \ \cos \xi \ \right]$$

$$= \sigma_{pv} \ \{1 + \left[(\underline{v}_{\perp LOS})/\underline{v}_I\right] \ |AR-1|\}$$

where

$\sigma_{vo}, \sigma_{pvo}$ = the dispersion of the reference target image along the major and minor axes of the radiation ellipsoid, parallel and perpendicular to the velocity vector.

σ_v, σ_{pv} = The current dispersions of the target image

ρ_o = reference range from the sensor to the target

ρ = current range

$\underline{v}_I$ = inertial velocity

$(\underline{v}_{\perp LOS})$ = projection of $\underline{v}_I$ onto the $\alpha - \beta$ plane

ξ = angle between the inertial velocity vector and the $\alpha - \beta$ plane, shown in Figure 3-8

$AR = \sigma_{vo}/\sigma_{pvo}$ - maximum aspect ratio of the reference image

The image resulting from the above equation is defined in the target frame coordinates. Both the coordinates of the hot spot centers and the dispersion matrix must be transformed into α - β plane coordinates [9:II-25]. From Figure 3-8

$$\cos \theta = \dot{\alpha}(t)/[\underline{v}_{\perp LOS}]$$

$$\sin \theta = \dot{\beta}(t)/[\underline{v}_{\perp LOS}]$$

where $(\underline{v}_{\perp LOS})$ is the magnitude of the velocity perpendicular to the tracker line of sight, defined as $[\dot{\alpha}(t)^2 + \dot{\beta}(t)^2]^{1/2}$. The hot spot coordinates are transformed to α - β coordinates by

$$\underline{x}_{\alpha\beta} = \begin{bmatrix} x \\ y \end{bmatrix}_{\alpha\beta} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{\text{target frame}} = \underline{A}\,\underline{x}$$

The dispersion matrix is transformed using

$$\underline{P}_{\alpha\beta} = \underline{A}\,\underline{P}\,\underline{A}^T$$

The inverse of $\underline{P}_{\alpha\beta}$ is required for Equation (3-8). This is conveniently obtained by using the equivalent expression

$$\underline{P}^{-1}_{\alpha\beta} = \underline{A}(\underline{P}^{-1})\underline{A}^T$$

These values are used in Equation (3-8) to calculate the intensity values for the measurement update.

3.7 Spatially Correlated Background Noise

The noise term, $v_{kl}(t_i)$, in Equation (3-9) contains spatially correlated background noise [3]. The correlation distance is about two pixels; the noise is modelled by non-zero circularly symmetric correlation between each pixel and its two closest neighbors in all directions.

The measurements are arrayed as a 64 element vector (64 pixels in the tracking window). The additive noise term is modeled as

$$\underline{v}(t_i) = \sqrt{\underline{R}}\,\underline{v}'(t_i)$$

where $\underline{v}'(t_i)$ is a vector of 64 independent, discrete-time zero-mean, white Gaussian noise processes with a variance of one. The strength of

the resulting noise process, $\underline{v}(t_i)$, is $E\{\underline{v}(t_i)\underline{v}^T(t_j)\} = \underline{R}\,\delta_{ij}$. This 64-by-64 matrix describes the spatial correlation between pixels and is presented in detail by Harnly and Jensen [3] and Kozemchak [4]. The correlation terms not involving first or second neighbors of a given pixel are essentially zero. The noise term added to the average intensity value of each pixel, forms the simulation measurement data. The adaptation for a wide-FOV is discussed in Section 4.4.

3.8 Summary

This chapter presents the truth model state propagation. Also included are the transformation from the deterministic inertial target trajectories to target motion on the FLIR image plane. Finally, the measurement model is developed. The next chapter describes the linear Kalman filter/correlator tracking algorithm and its use in a multiple model filter.

IV. TRACKING ALGORITHM

4.1 Introduction

The linear Kalman filter/correlator tracking algorithm that is the elemental unit in the multiple model adaptive filter (MMAF) was developed by Rogers [11] and Millner [7]. It uses a linear Kalman filter to provide estimates of target position, velocity, and acceleration and atmospheric disturbances. An enhanced correlation algorithm, using the estimated target shape as a template, provides the Kalman filter with pseudo-measurements. These pseudo-measurements are offset distances from the center of the FLIR FOV, modeled as a linear combination of the filter state variables. The linear formulation allows many terms of the Kalman filter to be precomputed, thus reducing on-line computational loading.

The equations necessary to propagate and update the state estimates are presented in this chapter. Additionally, the processing of the template and the correlation algorithm are described. The linear Kalman filter/correlator is the basic element in a bank of similar filters used by the MMAF, as discussed in Chapter II. The chapter concludes with the specific variations in each elemental filter used to define the MMAF.

4.2 State Space Model

The linear Kalman filter models target acceleration and atmospheric jitter position as stationary, first-order Gauss-Markov

processes. In the truth model, described in Chapter III, the atmospheric jitter position was represented by a third order model; however, the high-frequency double pole has little effect on the low frequency characteristics of the jitter and is neglected in the filter model [5].

The filter states are the target position (x_D, y_D), velocity (v_x, v_y), and acceleration (a_x, a_y), and jitter position (x_A, y_A), in each FLIR image plane direction.

$$\underline{x}_F = [x_D \; y_D \; v_x \; a_x \; a_y \; x_A \; y_A]^T \tag{4-1}$$

The state equations are

$$\begin{aligned}
\dot{x}_D &= v_x \\
\dot{y}_D &= v_y \\
\dot{v}_x &= a_x \\
\dot{v}_y &= a_y \\
\dot{a}_x &= (-1/\tau_{DF})a_x + w_{Dx} \\
\dot{a}_y &= (-1/\tau_{DF})a_y + w_{Dy} \\
\dot{x}_A &= (-1/\tau_{AF})x_A + w_{Ax} \\
\dot{y}_A &= (-1/\tau_{AF})y_A + w_{Ay}
\end{aligned}$$

where

τ_{DF} = correlation time for target acceleration

τ_{AF} = correlation time for atmospheric jitter

w_{Dx}, w_{Dy}, w_{Ax}, w_{Ay} = zero-mean white Gaussian noise processes whose strengths depend on tuning results

Identical, independent models are used to represent the effects in the x- and y-directions of the FLIR image plane.

The above relationships can be written as a state vector differential equation in standard form

$$\dot{\underline{x}}_F(t) = \underline{F}_F \underline{x}_F(t) + \underline{G}_F \underline{w}_F(t) \qquad (4\text{-}2)$$

where

$\underline{F}_F$ = The time invariant system plant matrix which is

$$\underline{F}_F = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/\tau_{DF} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1/\tau_{DF} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau_{AF} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau_{AF} \end{bmatrix}$$

$\underline{G}_F$ = the time invariant system noise output matrix

$$\underline{G}_F = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\underline{w}_F(t) = [w_{Dx}, w_{Dy}, w_{Ax}, w_{Ay}]^T$, the noise vector of mutually independent zero-mean white Gaussian noise processes

$$E\{\underline{w}_F(t)\underline{w}_F{}^T(t+\tau)\} = \underline{Q}_F \, \delta(\tau)$$

$$Q_{FD} = \begin{bmatrix} 2\sigma^2_{DF}/\tau_{DF} & 0 & 0 & 0 \\ 0 & 2\sigma^2_{DF}/\tau_{DF} & 0 & 0 \\ 0 & 0 & 2\sigma^2_{AF}/\tau_{AF} & 0 \\ 0 & 0 & 0 & 2\sigma^2_{AF}/\tau_{AF} \end{bmatrix}$$

σ^2_{DF} = target acceleration variance and ms value

σ^2_{AF} = atmospheric jitter position offset variance and ms value

The equivalent discrete time equation [5:173] is

$$\underline{x}_F(t_{i+1}) = \Phi_F(t_{i+1} - t_i)\underline{x}_F(t_i) + \sqrt{\underline{Q}_{FD}}\,\underline{w}_{FD}(t_i) \qquad (4\text{-}3)$$

where

$\underline{x}_F(t_i)$ = filter state vector at time, t_i

$\underline{w}_{FD}(t_i)$ = discrete-time zero-mean white Gaussian noise of covariance = $\underline{I}$

The state transition matrix is [13]:

$$\Phi_F(t_{i+1} - t_i) = \begin{bmatrix} 1 & 0 & t & 0 & J1 & 0 & 0 & 0 \\ 0 & 1 & 0 & t & 0 & J1 & 0 & 0 \\ 0 & 0 & 1 & 0 & J2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & J2 & 0 & 0 \\ 0 & 0 & 0 & 0 & J3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & J3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & J4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & J4 \end{bmatrix}$$

where

$$J1 = \tau_{DF}[\Delta t - \tau_{DF}(1 - \exp(-\Delta t/\tau_{DF}))]$$

$$J2 = \tau_{DF}[1 - \exp(-\Delta t/\tau_{DF})]$$

$$J3 = \exp(-\Delta t/\tau_{DF})$$

$$J4 = \exp(-\Delta t/\tau_{AF})$$

$$\Delta t = t_{i+1} - t_i$$

The growth in uncertainty due to dynamic driving noise is [8]:

$$\underline{Q}_{FD} = \begin{bmatrix} Q_{11} & 0 & Q_{13} & 0 & Q_{15} & 0 & 0 & 0 \\ 0 & Q_{11} & 0 & Q_{13} & 0 & Q_{15} & 0 & 0 \\ Q_{13} & 0 & Q_{33} & 0 & Q_{35} & 0 & 0 & 0 \\ 0 & Q_{13} & 0 & Q_{33} & 0 & Q_{35} & 0 & 0 \\ Q_{15} & 0 & Q_{35} & 0 & Q_{55} & 0 & 0 & 0 \\ 0 & Q_{15} & 0 & Q_{35} & 0 & Q_{55} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Q_{77} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_{77} \end{bmatrix}$$

where

$$Q_{11} = \sigma_{DF}^2 \left[2 \tau_{DF} \Delta t^3/3\right] - \left[2 \tau_{DF}^2 \Delta t^2\right] - \left[4 \tau_{DF}^3 \Delta t \exp(-\Delta t/T_{DF})\right] + \left[2 \tau_{DF}^3 \Delta t\right] - \left[T_{DF}^4 \exp(-2 \Delta t/\tau_{DF})\right] + \tau_{DF}^4$$

$$Q_{13} = \sigma_{DF}^2 \left[\tau_{DF} \Delta t^2\right] + \left[2 \tau_{DF}^2 \Delta t \exp(-\Delta t/\tau_{DF})\right] + \left[\tau_{DF}^3\right] - \left[2 \tau_{DF}^3 \exp(-\Delta t/\tau_{DF})\right] - \left[2 \tau_{DF}^2 \Delta t\right] + \left[\tau_{DF}^3 \exp(-2 \Delta t/\tau_{DF})\right]$$

$$Q_{15} = \sigma_{DF}^2 \left[-2 \tau_{DF} \Delta t \exp(-\Delta t/\tau_{DF})\right] + \left[\tau_{DF}^2\right] - \left[\tau_{DF}^2 \exp(-2\Delta t/T_{DF})\right]$$

$$Q_{33} = \sigma_{DF}^2 \left[2 \tau_{DF} \Delta t\right] - \left[3 \tau_{DF}^2\right] + \left[4 \tau_{DF}^2 \exp(-\Delta t/\tau_{DF})\right] - \left[\tau_{DF}^2 \exp(-2 \Delta t/\tau_{DF})\right]$$

$$Q_{35} = \sigma_{DF}^2 \{\tau_{DF} - \left[2 \tau_{DF} \exp(-\Delta t/\tau_{DF})\right] + \left[\tau_{DF} \exp(-2\Delta t/\tau_{DF})\right]\}$$

$$Q_{55} = \sigma_{DF}^2 \{1 - \exp(-2 \Delta t/\tau_{DF})\}$$

$$Q_{77} = \sigma_{AF}^2 \{1 - \exp(-2 \Delta t/\tau_{AF})\}$$

Using the terms defined above, the state estimate and covariance are propagated as follows:

$$\underline{\hat{x}}(t_{i+1}^-) = \Phi_F(t_{i+1}, t_i)\, \underline{\hat{x}}(t_i^+) \tag{4-4}$$

$$\underline{P}(t_{i+1}^-) = \Phi_F(t_{i+1}, t_i) \underline{P}(t_i^+) \Phi_F^T(t_{i+1}, t_i) + \underline{Q}_{FD} \tag{4-5}$$

$\underline{x}(t_i^+)$ = the state estimate after update at time t_i

$\hat{\underline{x}}(t_{i+1}^-)$ = the state estimate prior to update at time t_{i+1}

$\underline{P}(t_i^+)$ = the conditional covariance matrix after update at time t_i

$\underline{P}(t_{i+1}^-)$ = the conditional covariance matrix prior to update at time t_{i+1}

4.3 Measurement Model

The Kalman filter update process uses pseudo-measurements produced by an enhanced correlation algorithm. The template used by the correlator is developed by preprocessing the target image. The following subsections describe the target image processing, the correlation algorithm, and the Kalman filter update equations.

4.3.1 Two-Dimensional Fourier Transform. Many of the operations required to perform target image estimation are more easily accomplished in the Fourier domain. Additionally, Fourier transforms are readily accomplished optically; optical implementation of the tracking algorithm would greatly reduce the required computer resources. The Fourier transform of a complex-valued function of two independent variables, g(x,y), is a decomposition of g(x,y) into a linear combination of functions of the form $\exp[j2\pi(f_x x + f_y y)]$. The transform is defined as a double integral with respect to the spatial variables x and y, where f_x and f_y are spatial frequencies [12].

In this case, the discrete Fourier transform (DFT) is used because the FLIR measurements are discrete values of average intensity. The DFT and its inverse are:

$$H(f_x,f_y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} h(x,y) \exp[-j2\pi(f_x x + f_y y)]$$

$$h(x,y) = 1/N^2 \sum_{f_x=0}^{N-1} \sum_{f_y=0}^{N-1} H(f_x,f_y) \exp[+j2\pi(f_x x + f_y y)]$$

where

$H(f_x f_y)$ = transformed function in the spatial frequency domain

$h(x,y)$ = function in the spatial domain

f_x, f_y = spatial frequencies

x,y = spatial variables

N = the period of the assumed recurring sequence in both directions; thus the sequence of intensity values is discretized into an N x N pixel array.

The 8-by-8 tracking window is padded with additional data to form a 24-by-24 array for processing. The original array is padded to reduce edge effects, aliasing, and leakage conditions. Data, available from the larger field of FLIR data, is used for padding instead of zeros to prevent artificial edge effects [4:19]. The "large" FOV data is padded with zero for reasons discussed in Section 4.4.

4.3.2 Fourier Transform Shifting Property. In order to average the target image temporally, the intensity function in successive frames of data must be centered in the current FOV. The intensity function is centered on the filter's estimated target centroid location using the shifting property of the Fourier transform [9].

The Fourier transform shift theorem states that a linear phase shift in the frequency domain corresponds to a translation in the spatial domain. This phase shift can be thought of as a cylindrical shift due to the assumed periodic nature of the sampled data. Thus, using this property, the only difference between the centered image and a translated image is a linear phase shift proportional to the spatial displacement in the x- and y-directions. If $F[g(x,y)] = G(f_x,f_y)$, then

$$F[g(x-a,y-b)] = G(f_x,f_y) \exp[-j2 (f_x a + f_y b)]$$

where

a = spatial translation in the x-direction

b = spatial translation in the y-direction

As mentioned earlier, the filter's estimate of the target centroid location relative to the center of the FOV is used to determine the required phase shift to center the image for interframe shooting.

4.3.3 Exponential Smoothing. The available target image measurements are corrupted by FLIR and background noises. It is assumed that, for the applicable sample rates, these noises tend to vary more rapidly than the target intensity pattern from sample period to sample period [11].

This assumption is exploited by use of an exponential smoothing algorithm to time-average the data, thereby increasing the signal to noise ratio. Exponential smoothing approximates a true finite memory average without requiring the storage of a number of previous frames of data [6:33]. The applicable equation is

$$\hat{\underline{y}}(t) = \alpha \underline{y}(t) + (1-\alpha)\hat{\underline{y}}(t-1)$$

where

$\hat{\underline{y}}(t)$ = current average value

$\underline{y}(t)$ = current data frame

$\hat{\underline{y}}(t-1)$ = previous average data frame

α = smoothing constant, $0 < \alpha \leq 1$

The appropriate value of smoothing constant varies according to the dynamics of the target image variations. A rapidly changing image requires heavier weighting of more recent data and thus a higher value of α ; likewise, slower variations require smaller values. Based on previous studies, α is set at 0.1 for this effort [13].

The following steps summarize the template derivation process.

1. The Fourier transform of the raw FLIR data is calculated
2. The appropriate negating phase shift is applied to center the image based on the estimated target centroid location
3. Temporal smoothing of the centered data is performed
4. The intensity shape function is evaluated at the estimated state after the control application. For a single elemental filter

$$\underline{x}(t_{i+1}^{-c}) = \underline{x}_A(t_{i+1}^{-})$$

because, as stated in Section 1.3, control is applied so as to zero the predicted dynamic states. Superscript c denotes after application of the control. This is not the correct relation for the MMAF, as explained in Section 4.4.

4.3.4 Fast Fourier Transform (FFT) Correlation Algorithm. The correlator used in this effort is considered enhanced because it compares the raw data with a template, rather than the previous frame of data. The template is the estimated target intensity function located at the best estimate of the centroid offset, $\underline{x}(t_i^{-c})$. The FFT is used to perform the cross correlation by

$$\begin{aligned} F[\underline{g}(x,y)] &= \underline{G}(f_x f_y) \\ F[\underline{l}(x,y)] &= \underline{L}(f_x f_y) \\ [\underline{g}(x,y) * \underline{l}(x,y)] &= \underline{G}(f_x f_y) \cdot \underline{L}^*(f_x, f_y) \end{aligned} \tag{4-6}$$

where

$[\underline{g}(x,y) * \underline{l}(x,y)]$ = cross correlation of the two dimensional spatial sequences $\underline{g}(x,y)$ and $\underline{l}(x,y)$

$\underline{L}^*(f_x,f_y)$ = complex conjugate of the Fourier transform of the sequence $\underline{l}(x,y)$

The cross correlation, $\underline{R}(x,y)$, is obtained by taking the inverse FFT, or IFFT, of Equation (4-6).

The resulting cross correlation is subjected to a thresholding process to remove false peaks. Any element of $\underline{R}(x,y)$ that is less than the preselected fraction of the maximum correlation is set to zero. Following this, a centroid summation is used to locate the center of mass. It is assumed that the centroid of the thresholded correlation function is a good approximation of the peak location [11:53]; it also avoids the ambiguities that local peaks often cause peak-finding algorithms. The calculated centroid calculation is the

correlator's estimate of the offset of the target from the center of the data frame. These offsets are the "measurements" supplied to the linear Kalman filter.

4.3.5 The Kalman Filter Update Equations. The linear Kalman filter updates the state estimate based on the offset values from the correlator. The appropriate measurement equation is

$$\underline{z}(t_i) = \underline{H}_F \, \underline{x}_F(t_i) + \underline{v}_F(t_i) \tag{4-7}$$

where

$\underline{z}(t_i)$ = the offset estimate in x- and y-coordinates produced by the correlator, based on the filter predicted centroid location due to dynamics and atmospherics.

$$= \begin{bmatrix} x_{AD}(t_i) \\ y_{AD}(t_i) \end{bmatrix} + \begin{bmatrix} x_{AC}(t_i) \\ y_{AC}(t_i) \end{bmatrix} + \begin{bmatrix} v_{F1}(t_i) \\ v_{F2}(t_i) \end{bmatrix}$$

$$\underline{H}_F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$\underline{v}_F(t_i)$ = noise produced by the correlation algorithm with statistics that were shown empirically to be [11]

$$E\{\underline{v}_F(t_i)\} = \underline{0}$$

$$E\{\underline{v}_F(t_i)\underline{v}_F^T(t_j)\} = \underline{R}_F(t_i) \, \delta_{ij}$$

where

$$\underline{R}_F = \begin{bmatrix} .00436 & 0 \\ 0 & .00598 \end{bmatrix}$$

The standard Kalman filter update equations apply:

$$\underline{K}(t_i) = \underline{P}_F(t_i^-)\ \underline{H}_F^T[\underline{H}_F\ \underline{P}_F(t_i^-)\ \underline{H}_F^T + \underline{R}_F]^{-1} \tag{4-8}$$

$$\hat{\underline{x}}_F(t_i^+) = \hat{\underline{x}}_F(t_i^-) + \underline{K}(t_i)[\underline{z}(t_i) - \underline{H}_F\ \hat{\underline{x}}_F(t_i^-)] \tag{4-9}$$

$$\underline{P}_F(t_i^+) = \underline{P}_F(t_i^-) - \underline{K}(t_i)\ \underline{H}_F\ \underline{P}_F(t_i^-) \tag{4-10}$$

where all the terms have been previously defined.

4.4 Multiple Model Implementation

The theoretical basis for the multiple model adaptive filter is developed in Chapter II; this section concentrates on the variations in the elemental linear Kalman filter/correlators included in the MMAF bank of estimators. The single filter tracking algorithm discussed so far can potentially handle a target maneuvering at a level between ±20 g's, with tuning appropriate to that maneuver. By changing the values of τ_{DF} and σ_{DF}^2 used to model acceleration in the filter model, each filter in the bank can be tuned for different tracking scenarios. In order to establish a filter (or more than one) that can maintain lock during exceptionally harsh maneuvers, a larger tracking window is incorporated along with appropriate tuning. The larger 24-by-24 FOV is still processed as an 8-by-8 data array in an effort to limit computational loading. However, each "large" pixel is 60 μrads-by-60 μrads instead of 20 μrads on a side. This is accomplished by making each scalar intensity data value the average of 3-by-3 pixel array. The same FOV center is maintained for both the narrow and wide tracking windows.

The larger FOV requires additional modifications to the basic algorithm. The measurement noise for the large FOV is assumed to be spatially uncorrelated because the distance between "large" pixel centers is greater than the two "small" pixels that compose the spatial correlation distance. Also, the larger FOV makes it reasonable to assume the target intensity value at the edge of the tracking window is essentially zero; this makes it appropriate to pad the data array with zeros instead of noisy data, when estimating the target intensity function [13:IV-9]. Since the correlator output is in terms of pixels, "large" or "small", the pseudo-measurements for the large FOV case are multiplied by three, allowing the filter portion of the algorithm to remain unaffected by the different size tracking windows.

This effort evaluates three MMAFs; the first has two filters in the bank and the other two MMAFs add a third filter; one of these is a Bayesian MMAF and the other is a MAP MMAF formulation. The first elemental filter for each MMAF is tuned for a benign trajectory, using a narrow FOV for tracking accuracy. The other end of the expected maneuver range is covered by a wide-FOV filter tuned to track a 20 g target. In order to improve accuracy when the target motion is not well modelled by the first two filters, the small FOV third filter is tuned for 10 g maneuvers. The tuning results are presented in Table IV-1, below.

Table IV-1. Filter Tuning Values

		τ_{DF}(sec)	σ_{DF}^2 (pixels^2/sec^4)
MMAF 1	Filter 1	3.5	1000
	Filter 2	1.5	2000
MMAF 2	Filter 1	1.5	1000
	Filter 2	1.5	1500
	Filter 3	1.5	2000

The high value of σ_{DF}^2 for filter 1 in the first MMAF is due to the coarse discretization of the parameter space and is required to prevent MMAF 1 from diverging when tracking targets perform maneuvers at 20 g's.

Severe maneuvers can cause significant differences between the position estimates of the MMAF and the small FOV filters. These differences may become large enough to cause a complete cycle of the cylindrical shift, leading to divergence of the small FOV filter. Due to the artificial lower bounding, the Bayesian MMAF estimate continues to include these value, ultimately leading to divergence of the MMAF. Additionally, if a small FOV filter is allowed to diverge during harsh target maneuvering, it will not be capable of responding swiftly if the target motion again matches its internal dynamics model at a later time. To avoid this problem, a reacquisition process is initiated whenever the shift of the narrow tracking window exceeds a magnitude of 3.0 pixels. The divergent filter states are reset to a combination of the nondivergent filters' states or, for the MAP estimator, the MMAF state estimate. Likewise, for the

Bayesian MMAF, the covariance matrix is reset to the combined covariance values of the nondivergent elements; the MAP MMAF uses the current multiple model values for covariance. The conditional probabilities are left at the current values.

Earlier, it was pointed out that the estimated target intensity function is evaluated at the predicted target centroid position, after control application, to form the correlation template. With the MMAF, the control no longer zeros out the predicted offset of each elemental filter due to dynamics; instead the FLIR is reoriented to the predicted location of the target centroid due to target dynamics, based on the MMAF estimate. The intensity function is now evaluated at

$$[\hat{\underline{x}}_{DFk}(t_{i+1}^{-}) + \hat{\underline{x}}_{AFk}(t_{i+1}^{-})] - \hat{\underline{x}}_{DMMF}(t_{i+1}^{-})$$

for the k^{th} filter, where $\hat{\underline{x}}_{DFk}(t_{i+1}^{-})$ and $\hat{\underline{x}}_{AFk}(t_{i+1}^{-})$ are the k^{th} filter's target dynamic state and atmospheric jitter state estimates (predictions), respectively, and $\underline{x}_{DMMF}(t_{i+1}^{-})$ is the MMAF dynamics state estimate (prediction) based on the Bayesian or MAP criteria developed in Chapter II.

4.5 Summary

This chapter described the details of a tracking algorithm based on the multiple model adaptive filter developed in Chapter II. The first section presented the propagation equations for the basic linear Kalman filter. Next, the intensity function estimation

process used to develop the template for the enhanced correlator was outlined. The update equations completed the description of the linear Kalman filter/correlator which forms the basic element for the multiple model filter bank. The final section detailed the variations of this basic algorithm as implemented in the three MMAFs.

V. ALGORITHM TEST SET-UP

5.1 Introduction

This chapter presents the evaluation methods for the three tracking algorithms investigated during the course of this research. The first section covers the derivation of tracker statistics. The format of the performance plots, which are the major evaluation tool, is discussed next; followed by an outline of the parameter values used in the various test cases. The final section summarizes the three filters and defines the test cases used to generate the tracker performance statistics.

5.2 Tracker Statistics

Monte Carlo analysis techniques were employed to generate tracker performance statistics. Earlier studies established that ten runs allow sufficient convergence of sample error statistics to the true error statistics that would be observed with an infinite number of runs [7,13].

Tracking performance is determined by a filter's error in estimating the target's true position due to dynamics. Consistently low mean errors, along with low error variances, demonstrate the filter's ability to track the target through a wide range of maneuvers.

The sample mean errors of the filter position estimates are calculated using

$$\overline{E_{xd}}(t_i) = (1/N) \sum_{n=1}^{N} [x_{dn}(t_i) - \hat{x}_{dfn}(t_i)] \qquad (5\text{-}1)$$

$$= (1/N) \sum_{n=1}^{N} e_{xdn}(t_i)$$

where

$\overline{E_{xd}}(t_i)$ = sample mean error (averaged over N simulations) in x-dynamics position at time t_i

$x_{dn}(t_i)$ = truth model x-dynamics value at time t_i for run n

$x_{dfn}(t_i)$ = multiple model filter estimated x-dynamics value at time t_i for run n

$e_{xdn}(t_i)$ = number of Monte Carlo runs

The sample variance of the error is calculated by

$$\sigma_{xd}^2(t_i) = [1/(N-1)] \sum_{n=1}^{N} e_{xdn}^2(t_i) - [N/(N-1)] E_{xd}^2(t_i) \qquad (5\text{-}2)$$

where the terms are defined above. The same equations also apply to calculation involving the y-dynamics position. The errors are measured in FLIR image plane coordinates and represent offsets from the center of the sensor FOV. Error is expressed in units of pixels; each pixel is 20 μrads on a side.

Mean error and standard deviation are also time averaged in order to provide a scalar figure of merit for tracking accuracy. This temporal averaging occurs for a short period after transients have died out until just before the first maneuver, and from just after maneuver transients have declined until a time close to the end of the run. Time averaging allows presentation of data in a compact tabular form; however, these figures should be viewed in context.

Misleading figures of merit could occur; for example, if the error is ramping from positive to negative, then the average error could be zero while the actual error varies over a large range of values. Evaluation of the plots of mean error versus time provide a more accurate and complete representation of tracker performance.

5.3 Performance Plot Format

Performance plots are generated in order to evaluate fully the tracking algorithm's accuracy. These plots are of x- and y-dynamics estimate mean errors and mean error, plus and minus one standard deviation of the errors. The position estimate errors in pixels are plotted against time in seconds. Figure 5-1 is the y-position error before update and Figure 5-2 is the error after update. The maneuver takes place at t = 2.0 seconds; the plots show an increase in error followed by a recovery to approximately zero-mean tracking error. Notice that the maximum mean error after update is less than before update, properly showing that the update process reduces tracking error. Figure 5-3 compares the actual y-position rms error with the filter's estimate of its own rms error; this is a good indication of how well the filter is tuned. In this case, the maneuver initiation causes a short period of increased error; the filter responds appropriately by also increasing its computed rms error. This plot format is used to tune each elemental filter. In conjunction with the time averaged statistics, the mean ±1 standard deviation error time histories provide the means to evaluate the tracking algorithm's level of performance.

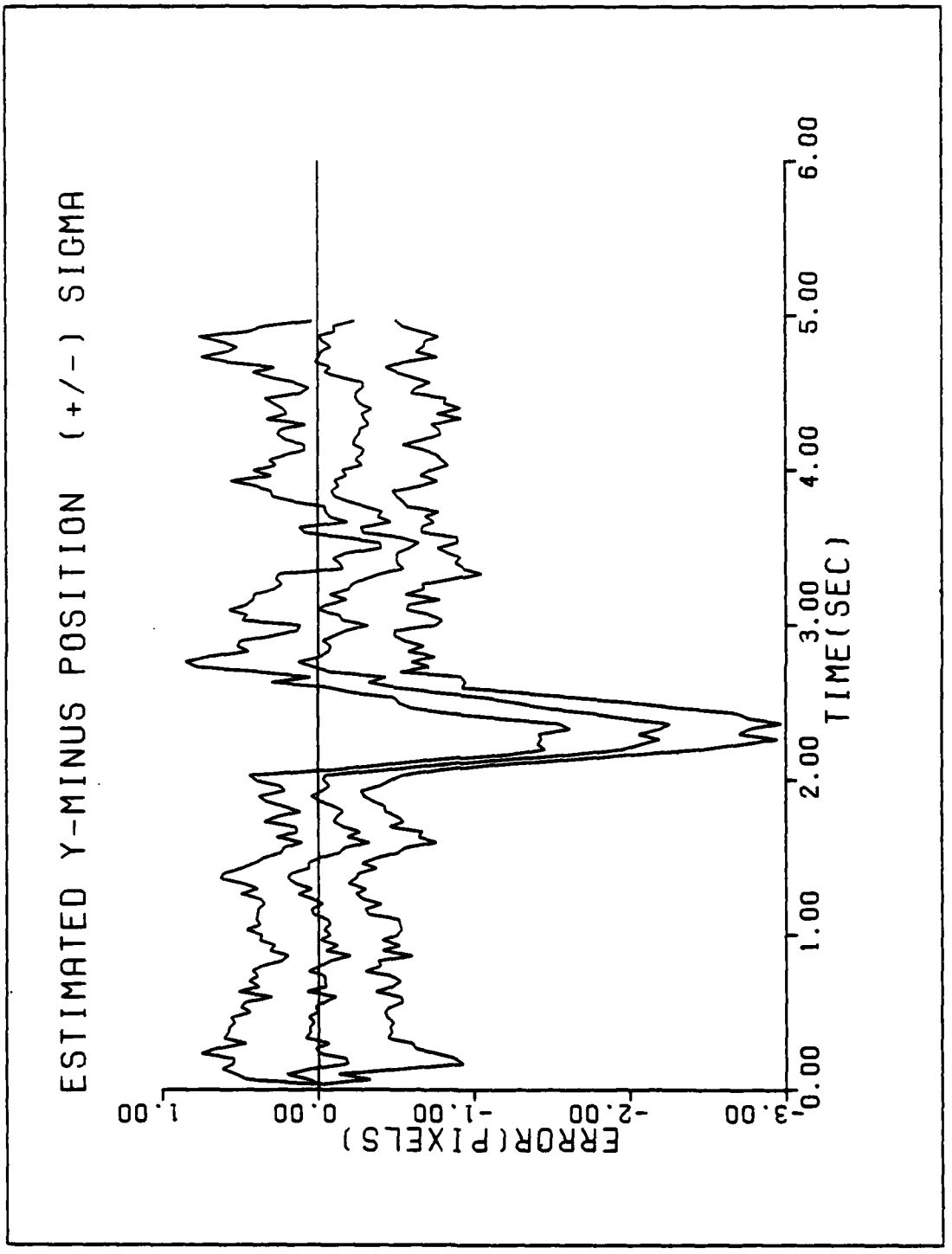


Figure 5-1. Typical Plot - Y-Dynamics Before Update

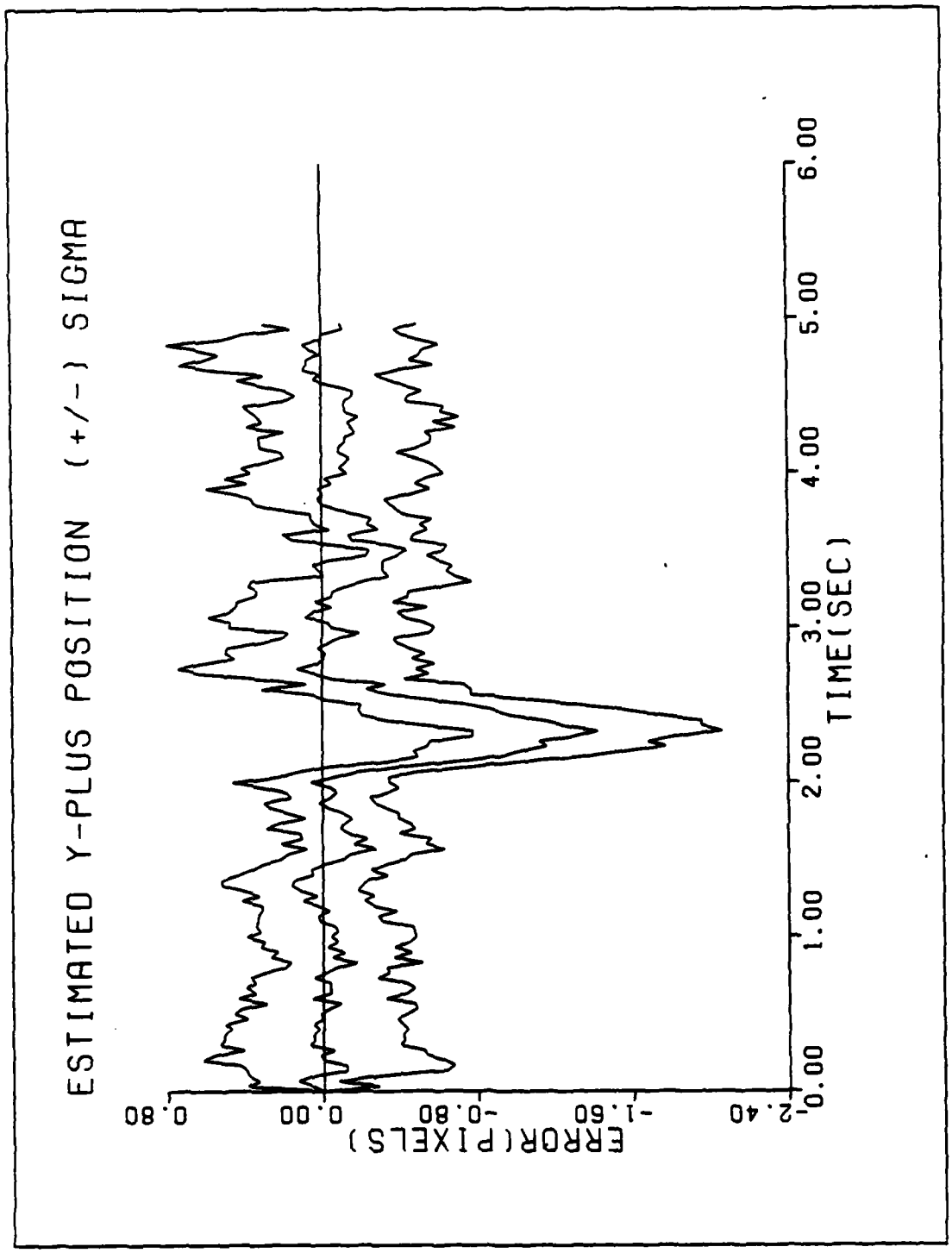


Figure 5-2. Typical Plot - Y-Dynamics After Update

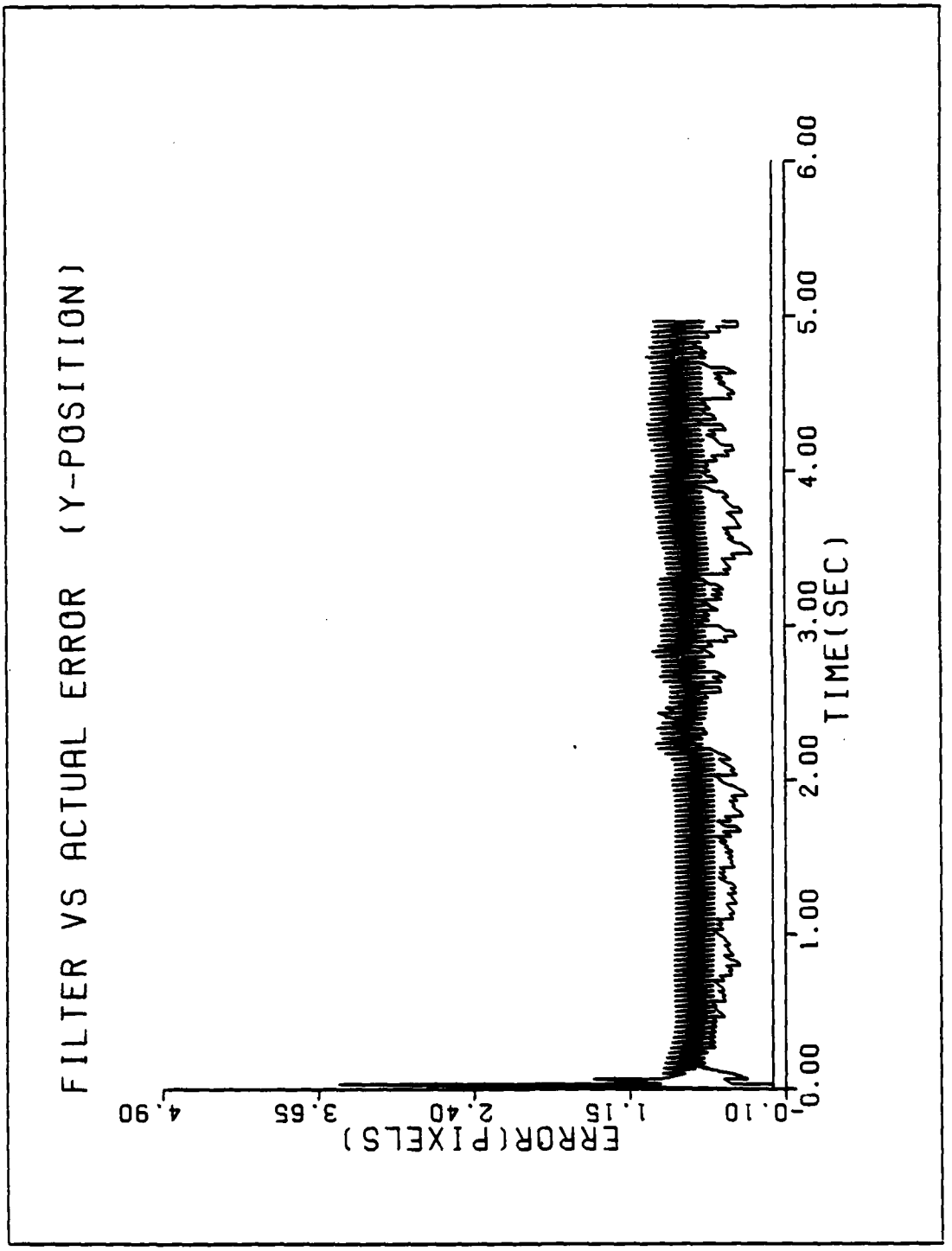


Figure 5-3. Typical Plot - Filter rms Error vs True rms Error

5.4 Parameter Values

The actual parameter values used in the equations developed in earlier chapters can significantly affect tracker performance. In all cases, the values used for this research are selected to produce the most accurate estimates of target position. Many of the values are the result of previous investigations [8,13]; however, some are due to the tuning work accomplished during this effort. Unless so stated, the following discussion applies equally to all three filters.

5.4.1 Truth Model Parameters. The initial conditions for all of the trajectories described in Section 3.2 are identical for every simulation:

Inertial position:	x =	5000 m
	y =	500 m
	z =	20000 m
Inertial velocity:	x =	-1000 m/s
	y =	0
	z =	0
Inertial acceleration:	x =	0
	y =	0
	z =	0

The maximum intensity of each hot spot, I_{max} of Equation (3-8) is 20. The variance of each $v_{kl}(t_i)$ in Equation (3-9), accounting for both FLIR and background noises, is one (intensity unit)^2. This leads to a signal-to-noise ratio of 20, which is representative of many tracking environments [3]. Atmospheric jitter variance, the variance of x_A in the diagram below Equation (3-1), is 0.2 (pixels)^2. Based

on a glint dispersion parameter of 2.0 $(\text{pixels})^2$, the aspect ratio, defined in Section 3.6, is one; i.e., the eigenvalues of $\underline{P}$ in Equation (3-8) are both two. These parameters applied to the model developed in Chapter III specify the simulation environment used to test the tracking algorithms.

5.4.2 Filter Parameter Values. For the MMAFs, each element filter was tuned to produce the best results at its individual design point without degrading the MMAF performance. In the two-element MMAF, the small FOV filter required a high dynamics noise variance to prevent MMAF divergence in the 20g scenarios and the large FOV filter is likewise detuned to maintain lock on a constant velocity target. However, the three-element MMAF has each filter tuned closer to the optimum values for the type of trajectory modeled by its internal dynamics model. The resulting tuning values are presented in Table IV-1.

For all the filters, an atmospheric correlation time (τ_{AF}) of 0.07 seconds is assumed. The filter atmospheric noise variance is 0.2 $(\text{pixels})^2$, like the truth model.

The probability lower bound is set at 0.001. The initial probability of filter one is 0.99 for the two element MMAF and 0.98 for the three-element MMAF. This high value indicates confidence that the target is essentially at constant velocity and not performing evasive maneuvers.

5.5 Test Scenario Descriptions

Although earlier chapters discuss every aspect of the tracking algorithms being investigated and the target test trajectories, a brief description of each will enhance understanding of the performance analysis contained in the next chapter.

5.5.1 Multiple Model Adaptive Filters. Three different tracking algorithms are being evaluated, the two-element baseline MMAF, the three-element Bayesian MMAF, and the three-element MAP MMAF. The baseline MMAF is taken directly from Suizu's work [13] based on the Millner algorithm [8] for the elemental filters. In addition to providing a basis for evaluating the three-element filters, the baseline MMAF is exercised for eight seconds (rather than five seconds) and operated at 100 Hz. These variations investigate crossover effects and the performance benefit of increased sample rate. The latter two MMAFs contain the same elemental filters; however, they create the MMAF estimate differently. The Bayesian MMAF produces the weighted average of the elemental filter's estimates, using the computed conditional probabilities as developed in Chapter II. Also using the conditional probabilities, the MAP MMAF chooses the elemental estimate with the highest probability of being correct. The performance of these three filters is evaluated over a number of test cases.

5.5.2 Test Cases. Each of the MMAFs are exercised against the same combination of trajectories at different acceleration levels, as shown in Table 5.1 below.

Table 5.1. Test Cases

		BASELINE	BAYESIAN	MAP
Traj 1 - straight line		X	X	X
Traj 2 - pull-up				
g level:	2	X	X	X
	10	X	X	X
	20	X	X	X
Traj 3 - pull-up/ constant velocity climb				
g level:	2	X	X	X
	10	X	X	X
	20	X	X	X

Trajectory four is not used because it is designed to investigate target image estimation rather than tracking accuracy. The target remains in approximately the same position in the FOV, changing only its range and orientation and does not stress the algorithm's ability to estimate target position. Additionally, each elemental filter is tested against trajectory two at the g level for which it is tuned. The elemental filter's performance with optimum tuning should represent the "best" tracking performance in that test case; because of purposeful detuning of various degrees, the elemental filter's performance may not be the "best". However, the comparison between the elemental filter and the MMAF gives some idea of the effect the multiple model approach has on tracking performance.

A test case designation code is useful to identify the different combinations of filters, trajectories, and g levels. The following code is used to identify the test cases in tables and plots.

T2	G20	F2-	BAY
1	2	3	4

1. Trajectory number

2. Maneuver g level (2,10,20)

3. Filter Type
   MM - multiple model
   F1, F2, F3 } element filters

4. Filter variation
   BAY - Bayesian MMAF
   MAP - MAP MMAF
   BASE - Baseline MMAF
   XT - Extended time, baseline MMAF
   100HZ - 100Hz sample rate, baseline MMAF

Under this scheme, the example code (T2G20F2-BAY) indicates the second elemental filter in the Bayesian MMAF operating in a 20g, trajectory two.

5.6 Summary

The evaluation methods presented in this chapter are used to describe the results obtained from operating the three MMAFs under various conditions. These conditions are described by the selected target trajectories and parameter values for the truth model and the filters. Based on the statistics and plot formats discussed in this chapter, the next chapter investigates the performance of the different MMAF formulations.

VI. PERFORMANCE ANALYSIS

6.1 Introduction

The performance analysis of the three MMAFs is discussed in this chapter. Each filter is evaluated separately, based on both its temporal averages and error statistics times histories. The baseline MMAF is presented first, along with the extended simulation (8 sec) and the 100 Hz sample rate version of this filter. The addition of a third elemental filter is investigated next by analyzing the Bayesian MMAF. Finally, the MAP MMAF results are evaluated. The concluding section compares and contrasts the results of all three MMAFs.

6.2 *The Baseline MMAF*

As discussed in Section 4.4, the narrow-FOV elemental filter requires a high level of dynamics driving noise to prevent total loss of target at 20 g's by the baseline MMAF. For all trajectories at low accelerations (1 - 2 g's), the conditional probabilities maintain a heavy weight (0.99+) on filter 1. This indicates that the actual residuals remain well matched to the internally computed ($\underline{H}\ \underline{P}\ \underline{H}^T + \underline{R}$) of that elemental filter. The high level of acceleration noise on the dynamic states allows this model to track a range of lower acceleration values rather than just zero acceleration.

In trajectory two, significant variations of the conditional probabilities occur at 10 and 20 g's. At 10 g's, the shift of weighting to filter 2 starts within 0.3 seconds after the maneuver is

initiated, and after two thirds of a second the weighting is heavily on filter 2. The MMAF responds slightly faster at 20 g's. No reacquisitions occur at 10 g's; however, the 20 g maneuver averages two reacquisitions per run for the narrow FOV filter. This is indicative of the narrow FOV filter having severe difficulty in maintaining lock when a 20 g step acceleration is introduced. Use of a more realistic acceleration model might remove the reacquisition requirement altogether. The conditional probabilities, following the maneuver, tend to oscillate in the 0.4 to 0.5 range and sometimes reach as high as 0.7. This is due to the purposeful detuning of the elemental filters to avoid total loss of the target at 20 g's; neither elemental filter produces residuals that are significantly large enough to push the weighting one way or the other. Up until the 3.5 second point in the simulation, trajectory three behavior is the same; however, at this point the maneuver change in this trajectory causes the wide FOV to be weighted much more heavily than the narrow FOV. Once again, as the target trajectory remains consistent after this step change in acceleration occurs, the MMAF begins to increase the weighting on filter 1.

The plots of mean error ± one standard deviation in the x-position, for trajectory two (Appendix A), do not show significant changes due to the maneuver, because the actual trajectory change in the x-direction is relatively small. The x-position mean error begins a positive ramp after the maneuver at 2.0 seconds; the maximum rate is 0.5 pixels/second during the 20 g maneuver. This ramping can possibly be attributed to the simplifying assumptions incorporated in

the models. First, each elemental filter assumes a non-rotating inertial reference frame when, in fact, the tracker-based coordinate frame does rotate. This model simplification can be mitigated by the addition of pseudo-noise to the position and velocity state estimates. Second, the true position is the location of the target center of mass, and the filter is tracking the target radiation centroid; usually the two are not coincident. A definitive explanation of the ramping has not yet been established, and this warrants further investigation.

Meanwhile, the y-position has a sharp increase in error following the maneuver, but quickly recovers (0.625 seconds) to zero-mean tracking error. The one sigma errors are slightly higher after the maneuver due to the increased emphasis on filter 2. The time averaged statistics in Table 6-1, show this and also, the tracking error is reduced following updates. Additionally, Table 6-1 indicates the effects of multiple model weighting on filter performance. Comparing the MMAF and filter 2 in a 20 g trajectory two, the single filter has consistently better performance. The same type of comparison for trajectory one indicates that filter 1 does not perform noticeably different from the MMAF, probably due to the consistently high weighting on filter 1 in the MMAF. Although it is expected that the tracking error will increase slightly for higher g levels, one notes that the MMAF actually performs worse in the well-modeled 20 g maneuver than in the 10 g maneuver that is not explicitly modeled by one of the elemental filters. This is further

evidence of the purposeful detuning of the elemental filters which leads to the oscillation in weighting of the two elemental filters discussed earlier in this section.

Table 6-1. Time Averaged Statistics (mean ± 1 σ) for the Baseline MMAF (3.5 - 5.0 sec)

CASE	x(-) (pixels)	x(+) (pixels)	y(-) (pixels)	(y+) (pixels)
T1MM-BASE	232 ±.424	-.219 ±.359	-.000 ±.394	-.003 ±.345
(T1F1-BASE)	.216 ±.426	.203 ±.377	-.009 ±.394	-.011 ±.345
T2G2MM-BASE	.214 ±.424	.198 ±.369	-.324 ±.404	-.008 ±.355
T2G10MM-BASE	.465 ±.431	.389 ±.374	-.120 ±.451	.008 ±.402
T2G20MM-BASE	1.29 ±.444	1.04 ±.388	-.179 ±.467	.044 ±.422
(T2G20F2-BASE)	.216 ±.420	.203 ±.365	-.009 ±.394	-.011 ±.345
T3G2MM-BASE	.127 ±.481	.158 ±.406	.181 ±.405	.131 ±.358
T3G10MM-BASE	.119 ±.516	.157 ±.429	.553 ±.506	.420 ±.441
T3G20MM-BASE	.305 ±.501	.376 ±.415	1.07 ±.514	.816 ±.441

The error statistics time history plots for x-position on trajectory three (Appendix A) are significantly different from trajectory two plots. At approximately 1.5 second into the simulation, the mean error experiences a step change, causing a bias until the maneuver at 3.5 seconds. The bias value is approximately 1.5 pixels for all acceleration levels; rms error levels remain about

the same. Returning to approximately zero-mean error at 3.5 seconds, the x-position plot demonstrates the characteristic ramping mean error as seen in trajectory two. There are no known differences between trajectories two and three in the first 3.5 seconds of the simulation to account for this unexpected behavior; it is supposed that the trajectory generation in the truth model may have an undiscovered variation.

Tracking performance in the y-direction is considerably better than the x-direction. This result is highly encouraging because tracking in the y-direction is significantly more difficult than in the x-direction; target maneuvers show the greatest change in y-direction motion. Furthermore, errors in position estimation along the length of the target are not as serious as errors orthogonal to the target longitudinal axis. The maneuver changes result in a maximum mean excursion of -5.25 pixels as 2.0 seconds, followed by a maximum of 2.0 pixels after the maneuver at 3.5 seconds. The filter recovers in approximately 0.75 seconds following the first maneuver; the response to the second change is somewhat slower, possibly due to the higher weighting of the wide FOV elemental filter. The time averages in Table 6-1 are artificially high because the averaging is performed from the 3.5 second maneuver point until simulation end; the plots also show a slow return to zero-mean tracking error.

6.2.1 Extended Time Baseline MMAF. The baseline MMAF, as implemented by Suizu [13], demonstrated a tendency to increase mean tracking error at approximately 5.0 seconds. This time represents the minimum range/maximum cross rate condition for the simulation; to

investigate tracking behavior beyond this condition, the simulation time was extended to 8.0 seconds. In most of the test cases, the extended runs demonstrated the same behavior as the baseline MMAF with a few notable exceptions.

The conditional probability behavior is virtually identical to the baseline MMAF for maneuvers of 1 - 10 g's. However, despite similar trends in system identification (e.g., appropriate heavy p_k values on the best matched filter), the extended runs show the MMAF losing lock in 20 g maneuvers. Unlike the same MMAF in a 5.0 second run, loss of lock occurs within 0.5 seconds of the maneuver at 2.0 seconds. This failure to maintain lock is attributed to a different noise time history caused by additional calls to the random number generator used to create the additive noise values for the additional frames of data in the extended runs. This loss of lock indicates that the tuning values are very marginal in the baseline MMAF.

The x-position mean ± one standard deviation error plots (Appendix D) for trajectory three differ significantly from the baseline MMAF plots. In this case, the unexplained step bias at 1.5 seconds is not observed. When the filter maintains lock, the y-position plots demonstrate recovery at the same rate as the baseline MMAF. These trends are further illustrated by the time average statistics in Table 6-2. As seen in the table, these statistics show slightly better mean tracking errors than the baseline, however, the values are averaged later in the simulation after the filter has more time to recover from maneuver transients.

Table 6-2. Time Averaged Statistics (mean ± 1σ) for the Extended Runs (6 - 8 sec)

CASE	x(-) (pixels)	x(+) (pixels)	y(-) (pixels)	(y+) (pixels)
T1MM-XT	.405 ±.436	.373 ±.385	.005 ±.426	.001 ±.378
T2G2MM-XT	.410 ±.433	.375 ±.385	-.020 ±.427	-.003 ±.377
T2G10MM-XT	.800 ±.459	.685 ±.412	.042 ±.424	.115 ±.373
T3G2MM-XT	.449 ±.432	.418 ±.382	.064 ±.418	.060 ±.369
T3G10MM-XT	.397 ±.429	.367 ±.377	.143 ±.419	.132 ±.370

6.2.2 Baseline MMAF - 100 Hz Sample Rate. Optical implementation of the tracking algorithm, as proposed by Roemer [10] and Rogers [11], allows data to be processed at the increased sample rate of 100 Hz. All other aspects remaining constant, additional updates should greatly improve tracking accuracy. In actuality, increasing the baseline 30 Hz sample rate to 100 Hz did not improve tracking performance. Under this condition, the MMAF diverged in every case and failed to maintain lock at higher (10 - 20 g's) acceleration levels.* This type of behavior may indicate problems in the measurement model. The increase in sample rate reduces the smoothing effect of the propagation portion of the algorithm, allowing mismodeled measurement data to have a more severe effect on the state estimate. A poor measurement model could lead the filter into erroneous evaluations of the incoming data causing poor tracking performance. Due to limitations on time and computer resources, this

* Subsequent investigation located a coding error. See Appendix E for updated results.

difficulty was not fully investigated; however, the plots which did not demonstrate loss of lock are included in Appendix E for completeness.

6.3 Bayesian MMAF

The Bayesian MMAF consists of the three elemental filters described in Section 4.4. This filter is tested against the same trajectories as the baseline MMAF. In all of the trajectories at low accelerations (1 - 2 g's), the rate of change of the conditional probabilities, following a maneuver, are very similar to the baseline MMAF; at the higher accelerations, the trends differ markedly. At 10 g's in trajectory two, following the maneuver, filter 2 is weighted more heavily than the other two filters, but within 1 second filter 1 is once again strongly weighted. In the 20 g maneuver, the weighting shifts to filter 3 for a few frames and then to filter 2, followed immediately by the narrow FOV filters (1 and 3) losing lock. After reacquisition, the conditional probability of filter 2 is very high (0.9+) and remains so until 1.3 seconds when filter 1 rapidly dominates the weighted average. In each case, less of the oscillations in probabilities previously observed in the baseline MMAF occurred. As observed in the baseline MMAF, when the filter obtains good acceleration estimates, the weighting returns to filter 1. Trajectory three also has the same conditional probability trends as observed in the baseline cases. However, following the maneuver change at 3.5 seconds, the filter is much slower in returning emphasis to filter 1; filter 1's probability, in a 10 g acceleration,

only reaches 0.68 as compared to 0.9+ in the baseline case, and the 20 case fails to maintain lock. Loss of lock occurs very quickly following the maneuver at 2.0 seconds; this is an additional unidentified peculiarity in trajectory three (in that it should yield the same characteristics as trajectory two until the 3.5 second point).

Tracking performance in the x-direction remains fairly consistent, including the unexplained bias observed only in trajectory three (Appendix B). The time averaged statistics in Table 6-3 show that the mean error tends to increase at higher accelerations, as does the error standard deviation. In most cases, the elemental filters, operating at the conditions for which they are specifically tuned, demonstrate lower mean errors and higher standard deviations than the MMAF in the same test case. This results from the MMAF mean value not being corrupted by the inclusion of means from mismatched filters; similarly, the covariance values are higher because the values in filters 2 and 3 are not reduced by being combined with the lower value used in filter 1.

Plots of error statistics time history demonstrate very good performance in the y-direction (Appendix B). In every case the MMAF recovers within 0.75 seconds following a maneuver; this recovery time is the same for the elemental filters. The elemental filter that is best tuned to the given maneuver (filter 2 or 3), alone, demonstrates lower maximum excursions following a maneuver than the MMAF.

Table 6-3. Time Averaged Statistics (mean ± 1σ) for the Bayesian MMAF (3.5 - 5.0 seconds)

CASE	x(-) (pixels)	x(+) (pixels)	y(-) (pixels)	(y+) (pixels)
T1MM-BAY	.228 ±.422	.215 ±.367	.026 ±.419	.025 ±.372
(T1F1-BAY)	.216 ±.420	.203 ±.365	.009 ±.394	-.012 ±.345
T2G2MM-BAY	.220 ±.429	.205 ±.374	-.012 ±.427	.013 ±.378
T2G10MM-BAY	.450 ±.456	.376 ±.400	-.239 ±.458	-.101 ±.406
(T2G10F3-BAY)	.388 ±.482	.321 ±.411	-.420 ±.429	-.245 ±.368
T2G20MM-BAY	1.13 ±.465	1.10 ±.411	-.271 ±.481	-.101 ±.449
(TG20F2-BAY)	1.02 ±.482	.793 ±.414	.062 ±.450	.331 ±.378
T3G2MM-BAY	.132 ±.507	.170 ±.422	.130 ±.434	.090 ±.376
T3G10MM-BAY	.088 ±.511	.129 ±.427	.367 1.440	.226 ±.340
T3G20MM-BAY	lost lock on target			

However, these two elemental filters demonstrate non-zero mean trends following recovery; filter 2 has a positively ramping bias and filter 3 never quite returns to zero-mean tracking error: it maintains a small negative bias until the end of the simulation. The MMAF does recover to zero-mean tracking; although in trajectory three at 20 g's, it recovers more slowly than at 10 g's, probably due to the longer emphasis on the wide FOV filter. Generally, the Bayesian MMAF outperforms the baseline MMAF; this is discussed in more detail in Section 6.5

6.4 MAP MMAF

This variation of the MMAF uses the output of the single "best" elemental filter to form the MMAF estimate. The selection is made based on the calculated conditional probabilities. In the lower acceleration cases, the MAP algorithm selects filter 1 exclusively and the actual calculated probabilities for filter 1 are higher than those observed for filter 1 in the Bayesian MMAF. Possibly, reacquisition, as discussed in Section 4.4, allows the diverging filters to be inappropriately weighted for a few frames, thus leading to different conditional probability weights subsequently. Following the 10 g maneuver in trajectory two, filter 2 is selected within 0.3 seconds; for the next second, the selection rotates through 3, 2, 1, and 3 in order. After 4.16 seconds, filter 1 is used for the remainder of the simulation. Similar behavior, about 2 sample periods sooner, is observed at 20 g's; the narrow FOV filters undergo reacquisition at about 2.5 seconds. Trajectory three, at both 10 and 20 g's, demonstrates the same trends until 3.5 seconds when the second maneuver change occurs. Following the return to constant velocity flight, the MAP MMAF begins to shift between filters 2 and 3. Filter 1 is selected again by 4.67 seconds, at 10 g's; however, at 20 g's, filters 2 and 3 continue to alternate until the end of the simulation. Thus, as observed previously, changes in acceleration trigger a shift to filters 2 and 3, while filter 1 is selected when a good estimate of constant acceleration is obtained [1].

Tracking performance in the x-direction is consistent with that in the other two MMAFs (Appendix C). Generally, as shown in Table 6-4, the MAP MMAF has lower mean errors and higher error standard deviations than the Bayesian MMAF. The ramp bias observed in both the baseline and the Bayesian MMAFs still occurs with approximately the same magnitude. Additionally, the unresolved step bias observed in trajectory three still remains; however, the bias magnitude is slightly higher, 1.6 and 1.9 pixels for 10 and 20 g's, respectively, as compared to 1.5 pixels in the first two MMAFs. The exclusive use of filter 1 data seems to make the output more sensitive to this phenomena.

Table 6-4. Time Averaged Statistics (mean ± 1σ) for the MAP MMAF (3.5 - 5 sec)

CASE	x(-) (pixels)	x(+) (pixels)	y(-) (pixels)	(y+) (pixels)
T1MM-MAP	.216 ±.420	.203 ±.366	.005 ±.416	.003 ±.350
T2G2MM-MAP	.216 ±.428	.198 ±.375	-.014 ±.432	.010 ±.387
T2G10MM-MAP	.453 ±.446	.377 ±.391	-.260 ±.528	-.129 ±.481
T2G10F3-MAP*	.400 ±.468	.333 ±.468	-.311 ±.608	-.126 ±.547
T2G20MM-MAP	1.26 ±.461	1.00 ±.412	-.344 ±.478	-.129 ±.454
T3G2MM-MAP	.109 ±.520	.149 ±.431	.136 ±.443	.095 ±.381
T3G10MM-MAP	.075 ±.512	.116 ±.427	.359 ±.454	.219 ±.383
T3G20MM-MAP	.159 ±.501	.247 ±.416	.991 ±.525	.697 ±.446

* The dynamics variances on target acceleration for the retuned filters 1, 2, and 3 are 300, 1900, and 2000 pixels^2/sec^4

As might be expected, the y-position error statistics plots (Appendix C) demonstrate the effects of MAP estimation more clearly. Low acceleration (0 - 2 g's) maneuvers are tracked entirely by filter 1. The maximum excursion of mean error observed is 0.5 pixels and the error standard deviation remains under 0.5 pixels. In this case, accuracy has been sacrificed by the purposeful detuning of filter 1, required for the MMAF to maintain lock in 20 g maneuvers. Trajectory two shows a slight tendency toward a negative bias following recovery from the maneuver. At 10 and 20 g's, the maximum excursions are -1.6 and -4.0 pixels and the recover time is 0.68 and 0.75 seconds, respectively. The increased standard deviation due to the selection of filter 2 or 3, following the maneuver, is very noticeable in the 10 g case and less so, due to scaling, in the 20 g case. In an effort to evaluate the effect of detuning, the elemental filters are tuned more tightly, using the values noted in Table 6-4 rather than those established in Section 4.4. The MAP MMAF is tested at 10 g's (loss of lock by the MMAF at 20 g's is the only reason for detuning the elemental filters); the MMAF with this tuning has a slightly smaller maximum excursion (-1.5 pixels) and quicker recovery (0.625 seconds), but it shows a greater tendency towards a negative bias following recovery (Appendix C). The time averaged statistics, in Table 6-4, indicate that most of the improvement is in the x-direction tracking rather than the y-direction which, as discussed earlier, is not the difficult direction to track. In this light, the current tuning values appear more favorable than might be anticipated. Trajectory three performance also demonstrates these

trends. The slightly higher standard deviations following maneuvers are observable but not obvious in the plots. The MAP estimates are not significantly lower in this case than the other tow filters; this appears to be a consequence of the cycling between filters which leads to an effective averaging of the elemental filter's outputs.

6.5 Comparison of the Three MMAFs

A comparison of the three different MMAFs (baseline, Bayesian, and MAP) allows one to evaluate the effects of the third filter in the bank and the MAP estimation approach versus the Bayesian approach. Several aspects of each filter are used to judge relative performance. The maximum excursion of the mean error following a maneuver indicates how well the MMAFs handle changing target motion. Additionally, the time to recover to zero mean tracking error after the maneuver is another measure of performance in this category. Observation of the time averaged statistics yields an idea of each MMAF's accuracy in steady state.

Clearly, as seen in Table 6-5, the addition of the third filter to the bank reduces the maximum mean error following the maneuvers at 2.0 and 3.5 seconds. The differences between the Bayesian and MAP MMAFs do not indicate a significant difference between the two approaches; however, the MAP MMAF may handle 20 g maneuvers better. The evidence is inconclusive due to the difficulties with trajectory three discussed earlier in this chapter.

Table 6-5. Maximum Mean Error Excursion Following Maneuvers at 2.0 and 3.5 Seconds

	Trajectory Two (2.0 sec)		Trajectory Three (2.0/3/5 sec)	
	10 g's (pixels)	20 g's (pixels)	10 g's (pixels)	20 g's (pixels)
Baseline	-2.38	-6.5	-2.38/1.25	-6.5/2.0
Bayesian	-1.4	-4.13	-1.51/1.18	lost lock
MAP	-1.6	-4.00	-1.51/1.11	-5.38/2.5

Response time after a maneuver (Table 6-6) doesn't vary significantly between the three MMAFs. Immediately following a maneuver, when the conditional probabilities shift, it takes about the same number of sample periods to cause a change of filter in the MAP MMAF as to achieve a 0.9+ weighting of the same filter in the Bayesian MMAF.

Table 6-6. Recovery Time After Maneuvers at 2.0 and 3.5 Seconds

	Trajectory Two (2.0 sec)		Trajectory Three (2.0/3/5 sec)	
	10 g's (sec)	20 g's (sec)	10 g's (sec)	20 g's (sec)
Baseline	.625	.75	.688/ *	.75/ *
Bayesian	.75	.75	.688/.75	lost lock
MAP	.688	.75	.688/.688	.813/.938

* Never reached zero-mean tracking error in the 5.0 sec

The time averaged statistics shown in Table 6-7 do not indicates any overwhelming trends. There is no strong evidence that the addition of the third filter improves steady state tracking error in trajectory two; however, trajectory three, y-direction, demonstrates reduced mean errors except when the Bayesian MMAF loses lock. The MAP MMAF does demonstrate generally lower means and higher standard deviations than the Bayesian MMAF, but not significantly. Notice that the standard deviations remain consistently between about 0.35 and 0.45; this is very important for insuring that the laser energy "paints" a very small area of the target.

Table 6-7. Time Averaged Statistics (mean ± 1σ) After Update

	Trajectory Two		Trajectory Three	
	x(+) (pixels)	y(+) (pixels)	x(+) (pixels)	y(+) (pixels)
2 g's				
Baseline	.198 ±.369	.008 ±.355	.158 ±.406	.131 ±.358
Bayesian	.205 ±.374	.013 ±.378	.170 ±.422	.090 ±.376
MAP	.198 ±.375	.010 ±.387	.149 ±.431	.095 ±.381
10 g's				
Baseline	.389 ±.374	.008 ±.422	.157 ±.429	.420 ±.441
Bayesian	-.376 ±.400	-.101 ±.406	.129 ±.427	-.226 ±.340
MAP	.377 ±.391	-.129 ±.481	.116 ±.427	.219 ±.383
20 g's				
Baseline	1.04 ±.376	.044 ±.422	.376 ±.415	.816 ±.441
Bayesian	-1.13 ±.456	.065 ±.449	lost lock	
MAP	1.00 ±.412	-.129 ±.457	.247 ±.416	.697 ±.446

VII. Conclusions and Recommendations

7.1 Conclusions

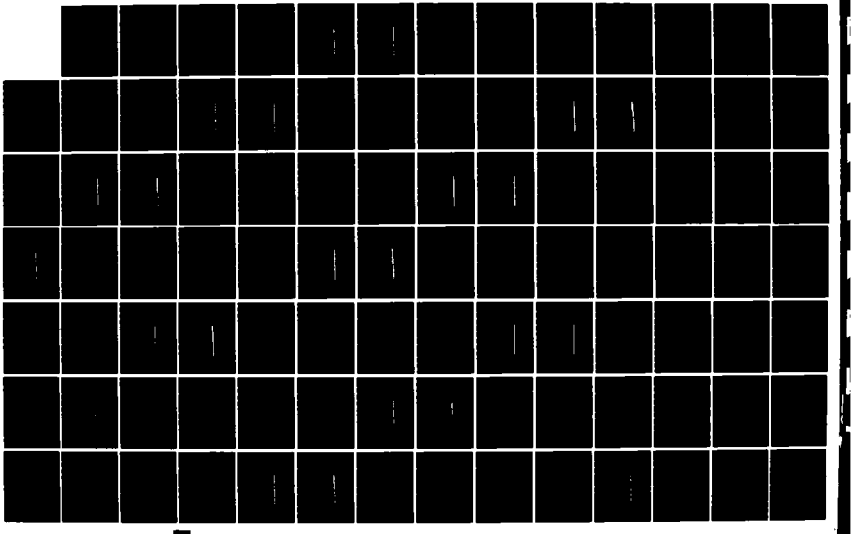
The ability to track a highly dynamic target is enhanced by the addition of a third elemental filter to the multiple model adaptive filter bank. Lower peak mean errors increase the likelihood of achieving a kill with a laser weapon system. The additional computational loading required by a third element becomes almost insignificant if parallel processing techniques are employed.

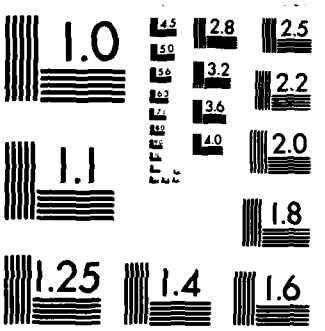
The baseline MMAF does not encounter any difficulty with the minimum range/maximum crossing rate condition; however, it demonstrates divergent characteristics at a 100 Hz sample rate. This poor performance at an increased sample rate indicates a need to question the basic measurement model employed in each elemental filter.

Initially, it was expected that the differences between Bayesian and MAP estimation would be more noticeable. The anticipated faster response of the MAP MMAF to maneuvers was not demonstrated. Average mean tracking errors and the maximum mean excursions were very similar in both approaches. The inability of the algorithms to maintain consistent system identification, e.g., maintain high conditional probabilities on the "best" matched model, could explain these ambiguous results. Additionally, the reacquisition function may have prevented more accurate and consistent system identification; however, because it only appeared during 20 g maneuvers, it is a relatively minor influence. Overall, the study

AD-A155 466 BAYESIAN VS MAP MULTIPLE MODEL ADAPTIVE ESTIMATION FOR FIELD OF VIEW EXPA..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. P A LOVING 2/4

UNCLASSIFIED MAR 85 AFIT/GE/EE/85M-1 F/G 12/1 NL





shows that an additional element in the MMAF bank improves performance and that both Bayesian and MAP estimation techniques support accurate tracking of highly dynamic airborne targets.

7.2 Recommendations

Further investigation of the multiple model adaptive filter to resolve the difficulties encountered and expand current knowledge is recommended. Future efforts should address the following areas:

- Development of more realistic target trajectories. Specifically, the step changes in accelerations should be replaced with models that more closely approximate typical target behavior. This may allow the MMAF to maintain lock with tighter tuning of the elemental filters.
- Investigate the step bias in x-position observed only in trajectory three
- Test the tracking algorithms against targets with varying shape functions caused by changes in target orientation and variations in size, shape, and number of hot spots
- Indepth investigation of the 100 Hz sample rate, concentrating on possible inaccuracies in the measurement model. By using a single elemental filter, computational loading can be limited; additionally, there would be no question whether the multiple model formulation is the cause of this behavior.
- Implementation of the constant turn rate target acceleration model (implemented by Suizu [13] in the extended Kalman filter formulation) for one or more of the elemental filters. This model more closely represents high performance aircraft evasive actions, particularly at short range.
- Change hot spot separations to values more representative of real world targets
- Add pseudo-noise to the position and velocity states to account for the rotating reference frame of the tracker
- Replace the current impulsive control with a more realistic control response

- Improve the reacquisition function by preventing the data from the diverging filters from being used for several sample periods following reacquisition. Currently, if the MMAF criteria (Bayesian or MAP) selects a diverging filter following reacquisition, it is included even though its estimate is degraded. By preventing the use of these estimates for a few sample periods, the conditional probabilities could again converge to appropriate values.

- Implement the algorithm on a small word-length machine

- Investigate the implications of tracking the radiation centroid, which may not be located on the target, rather than a physical point on the target. Include the effects of the hot-spot produced by laser illumination of the target on tracking performance.

APPENDIX A

Performance Plots for the Baseline MMAF

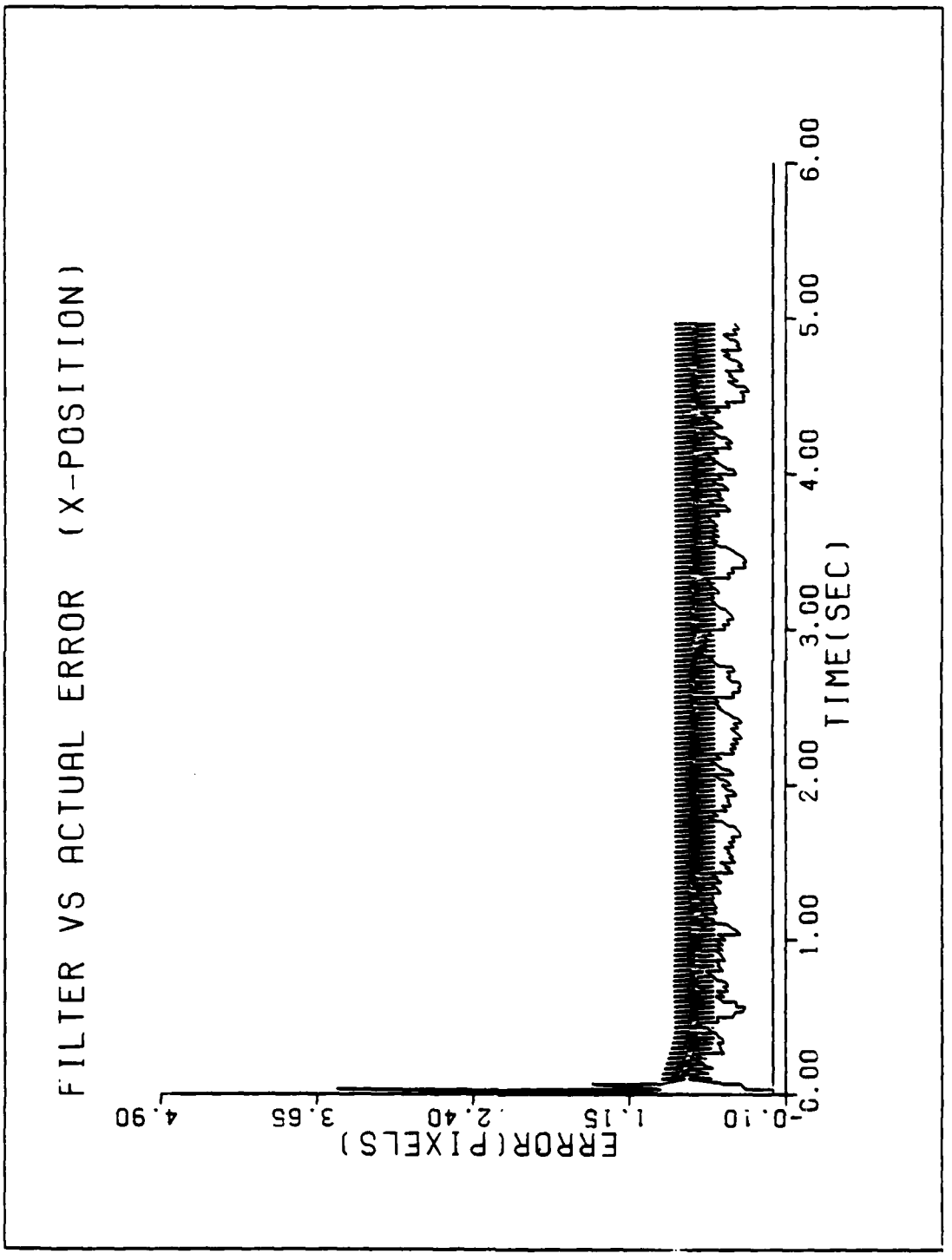


Figure A-1a. Performance Plot for TIMI-BASE

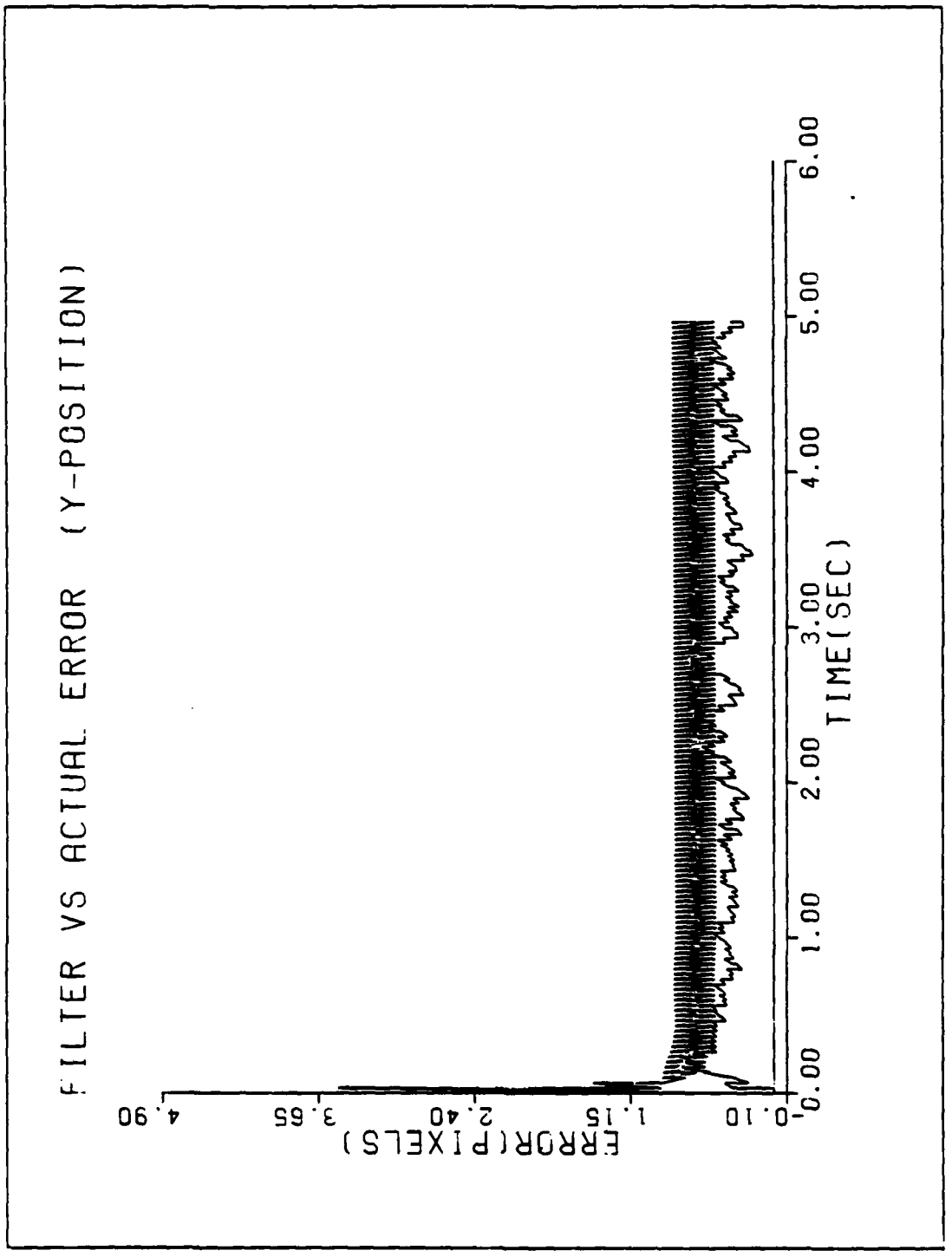


Figure A-1b. Performance Plot for TEMM-BASE

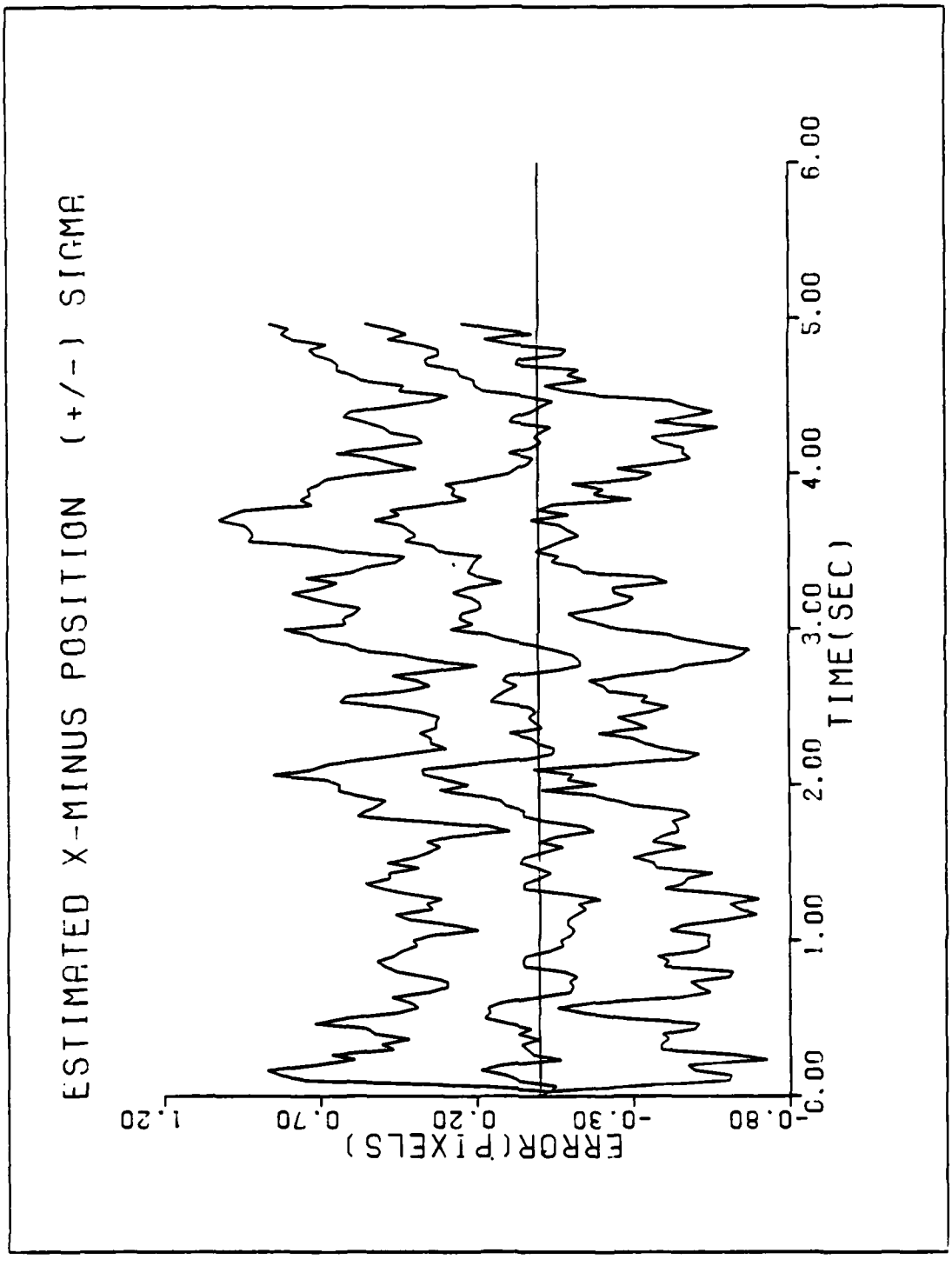


Figure A-1c. Performance Plot for TIME-BASE

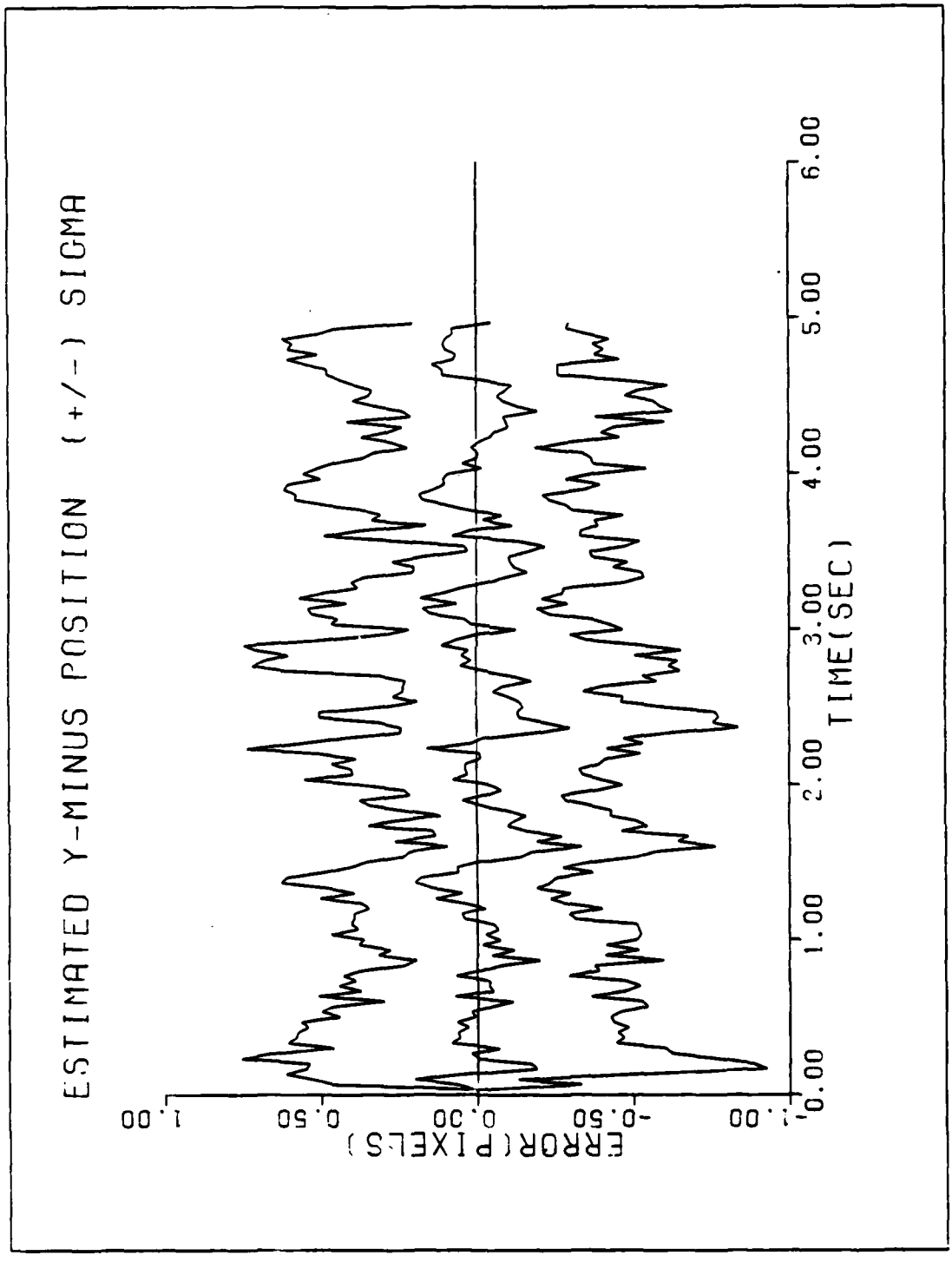


Figure A-1d. Performance Plot for TEST-BASE

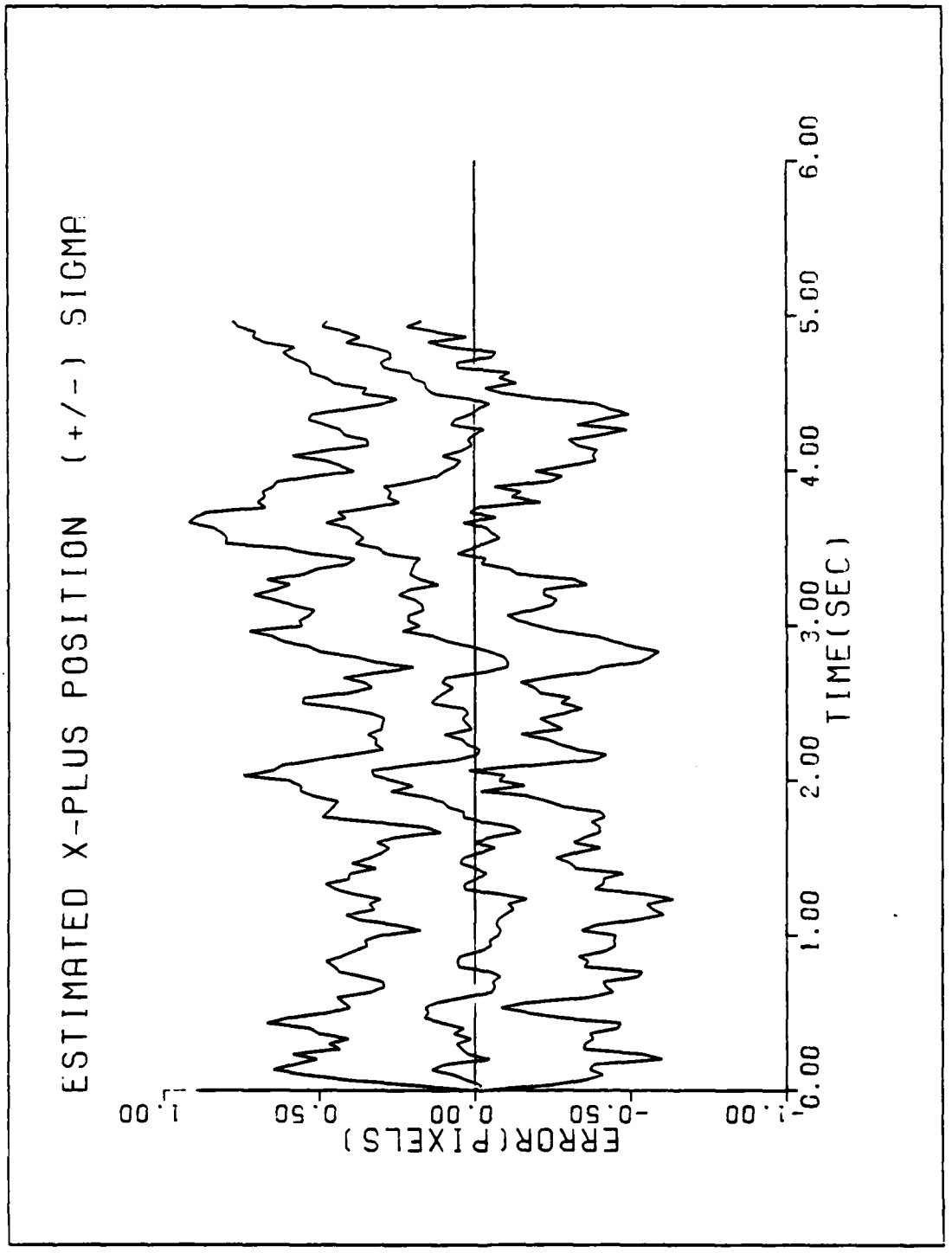


Figure A-1e. Performance Plot for TIME-BASE

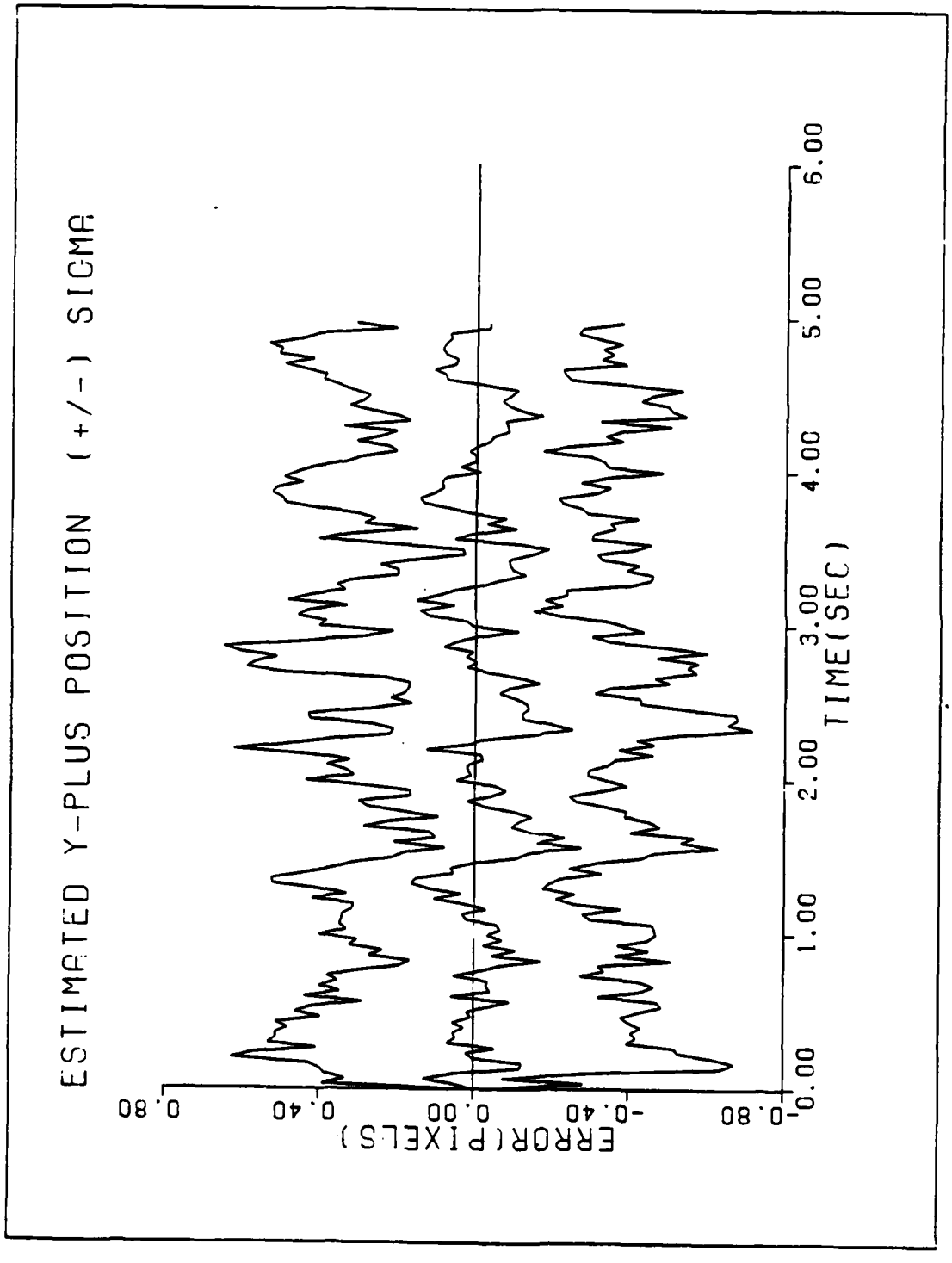


Figure A-1f. Performance Plot for TIMM-BASE

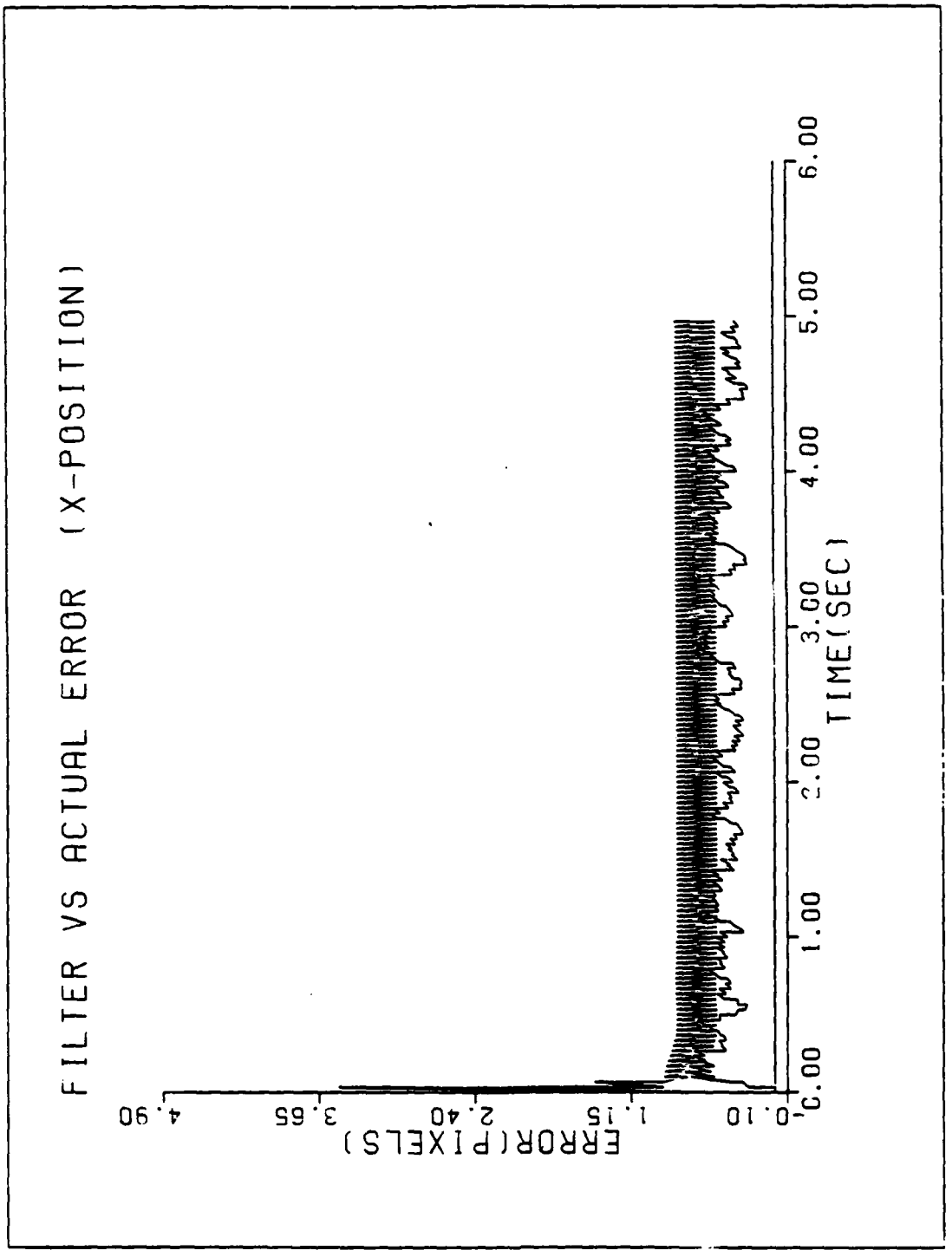


Figure A-2a. Performance Plot for TIFI-BASE

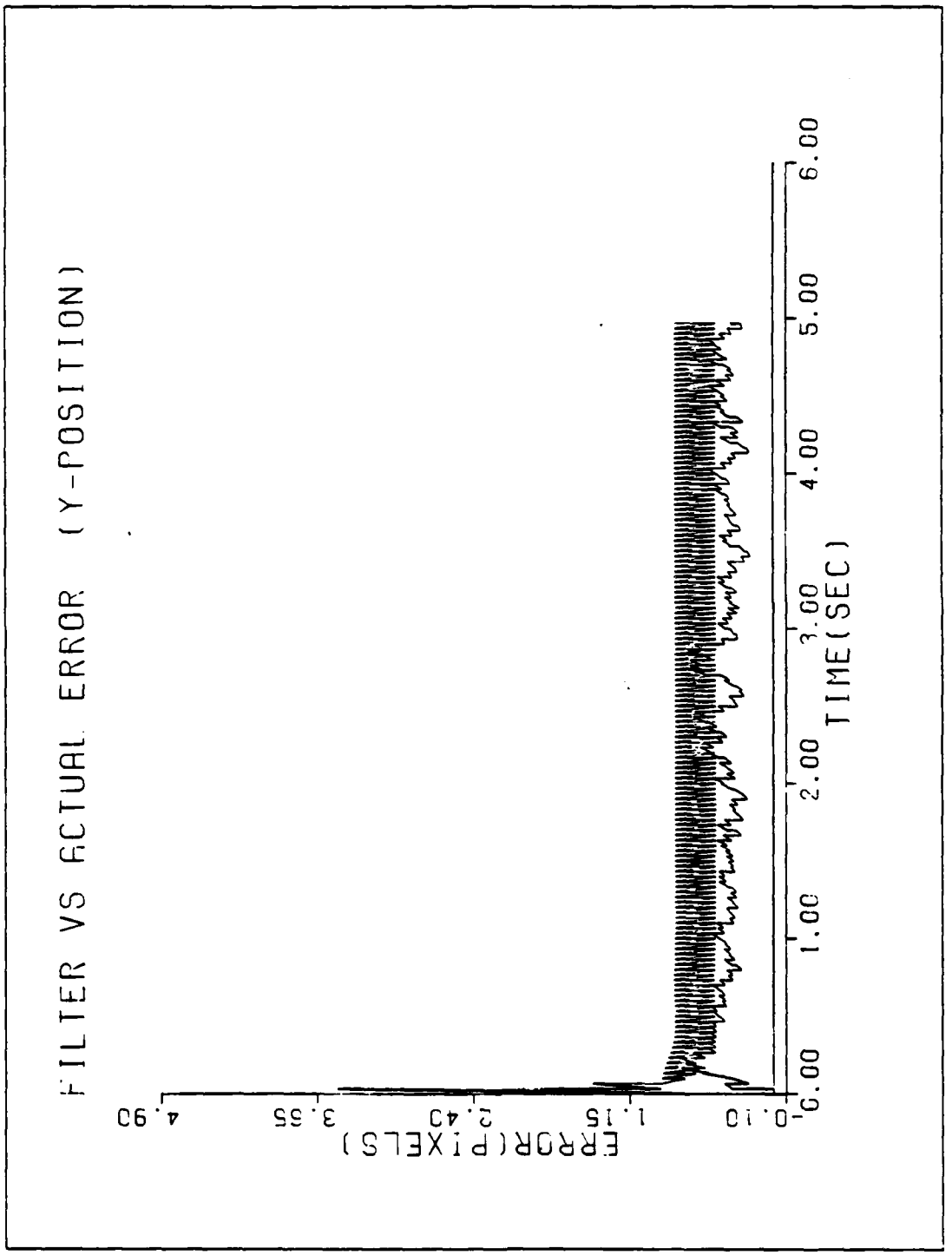


Figure A-2b. Performance Plot for TIFI-BASE

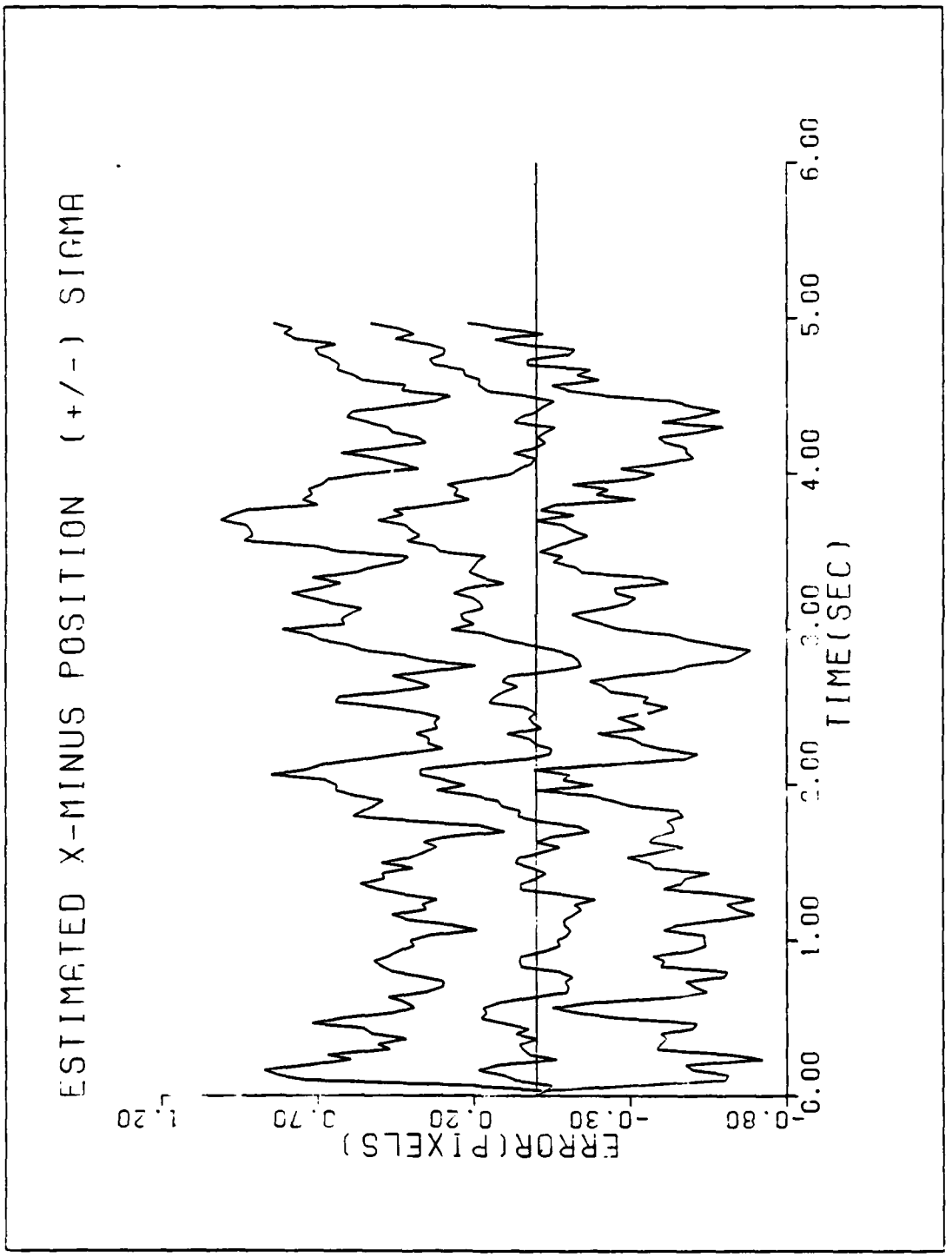


Figure A-2c. Performance Plot for TIFI-BASE

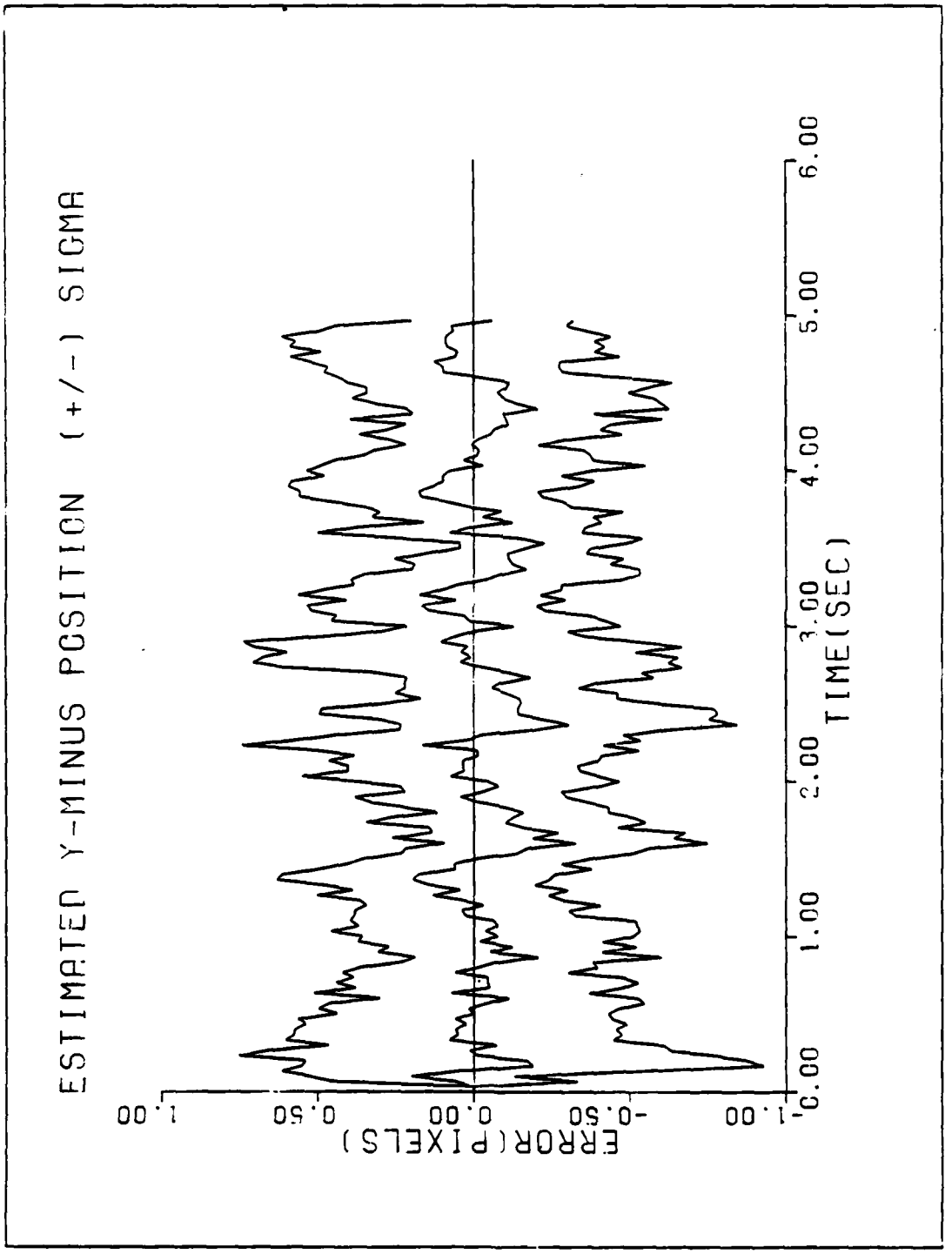


Figure A-2d. Performance Plot for TIFI-BASE

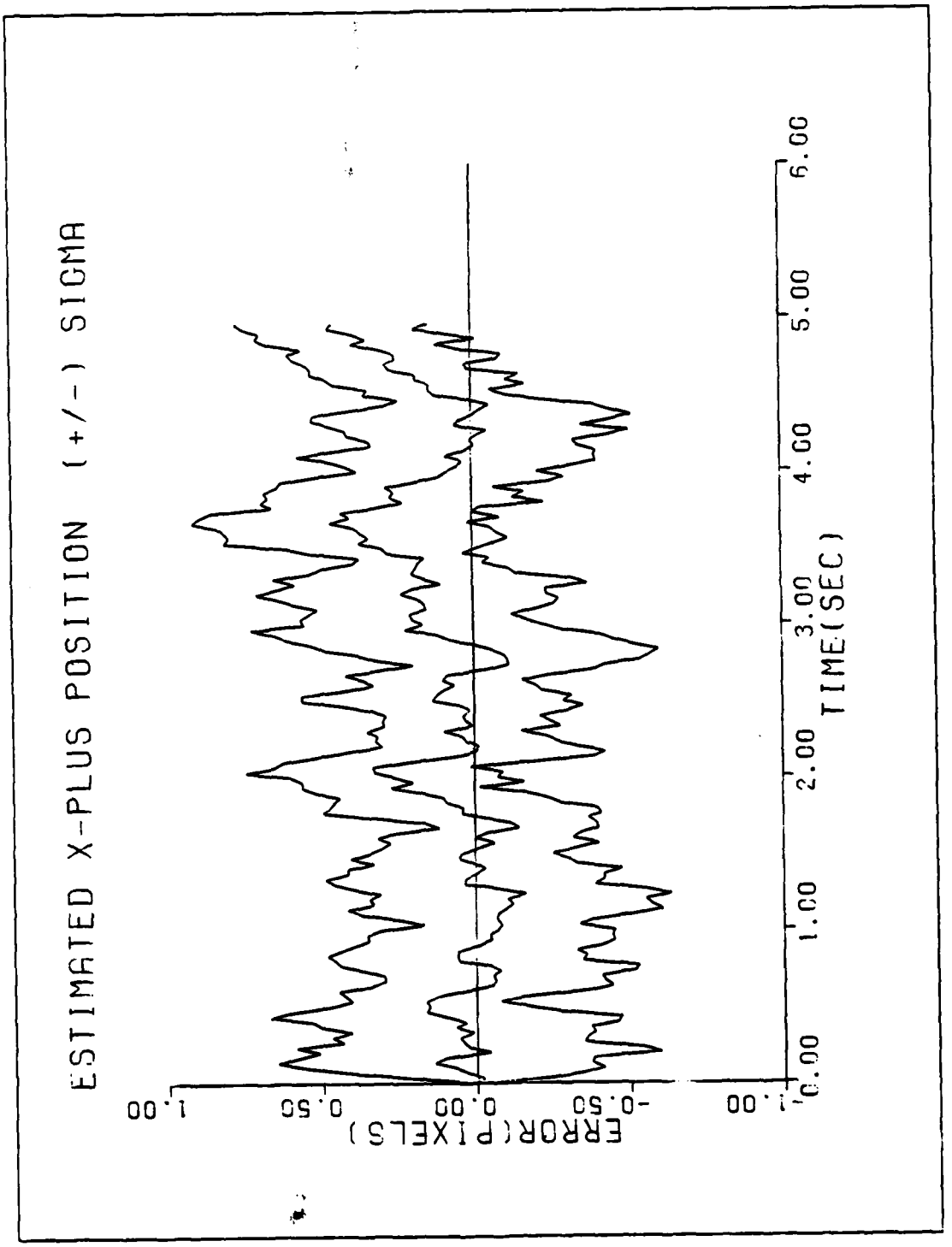


Figure A-2e. Performance Plot for TIFI-BASE

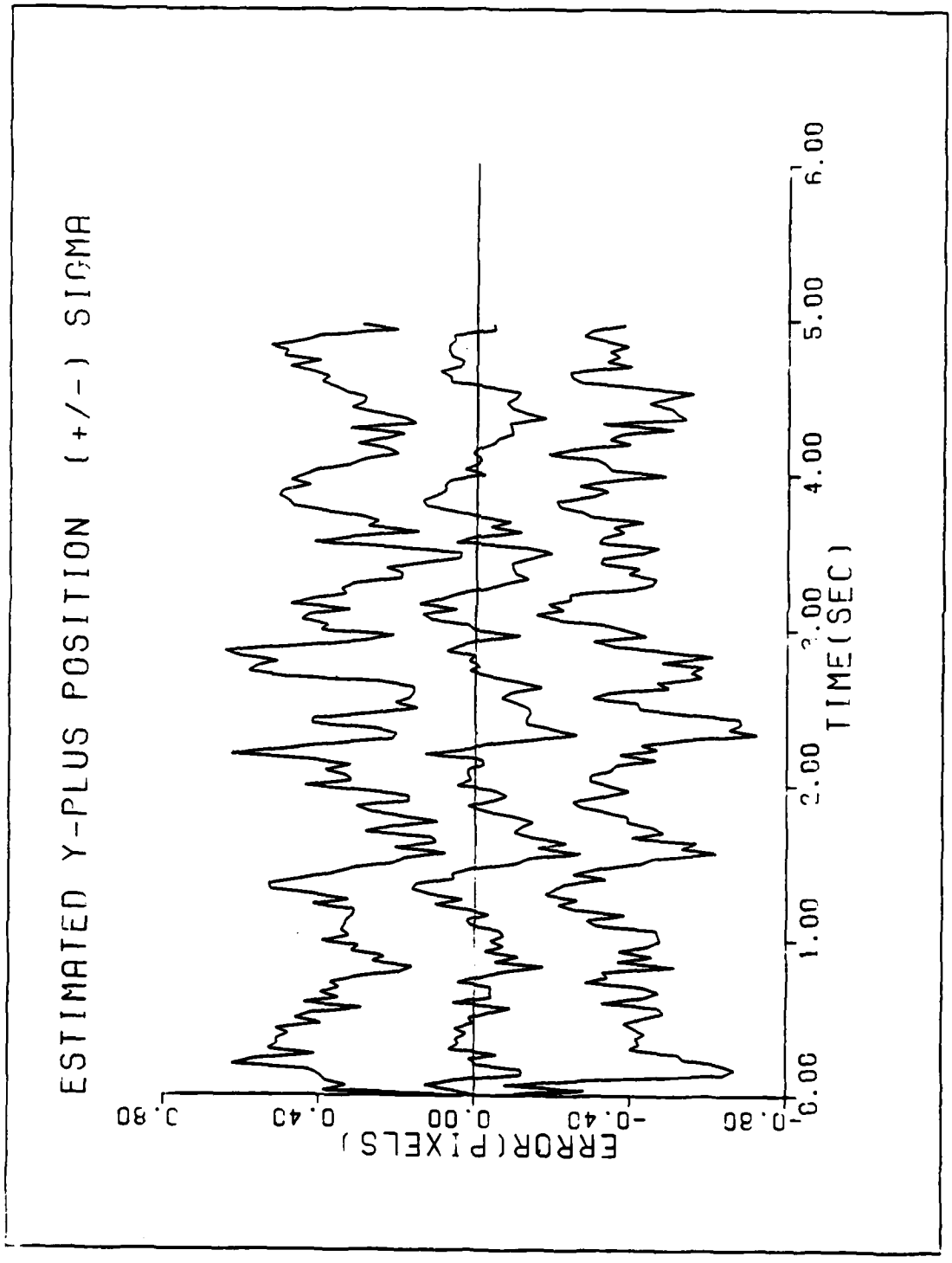


Figure A-2f. Performance Plot for TIFI-BASE

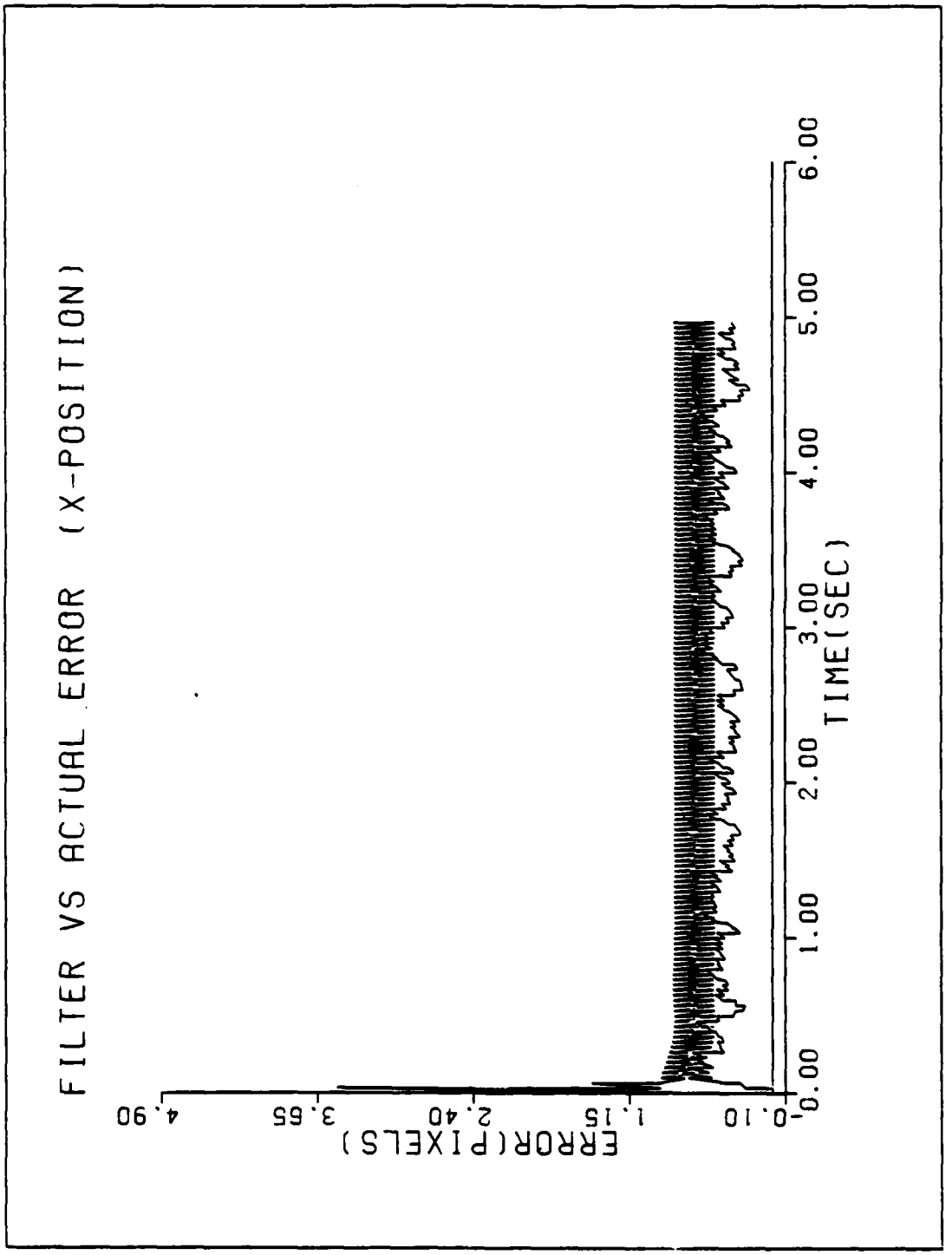


Figure A-3a. Performance Plot for T2G2MM-BASE

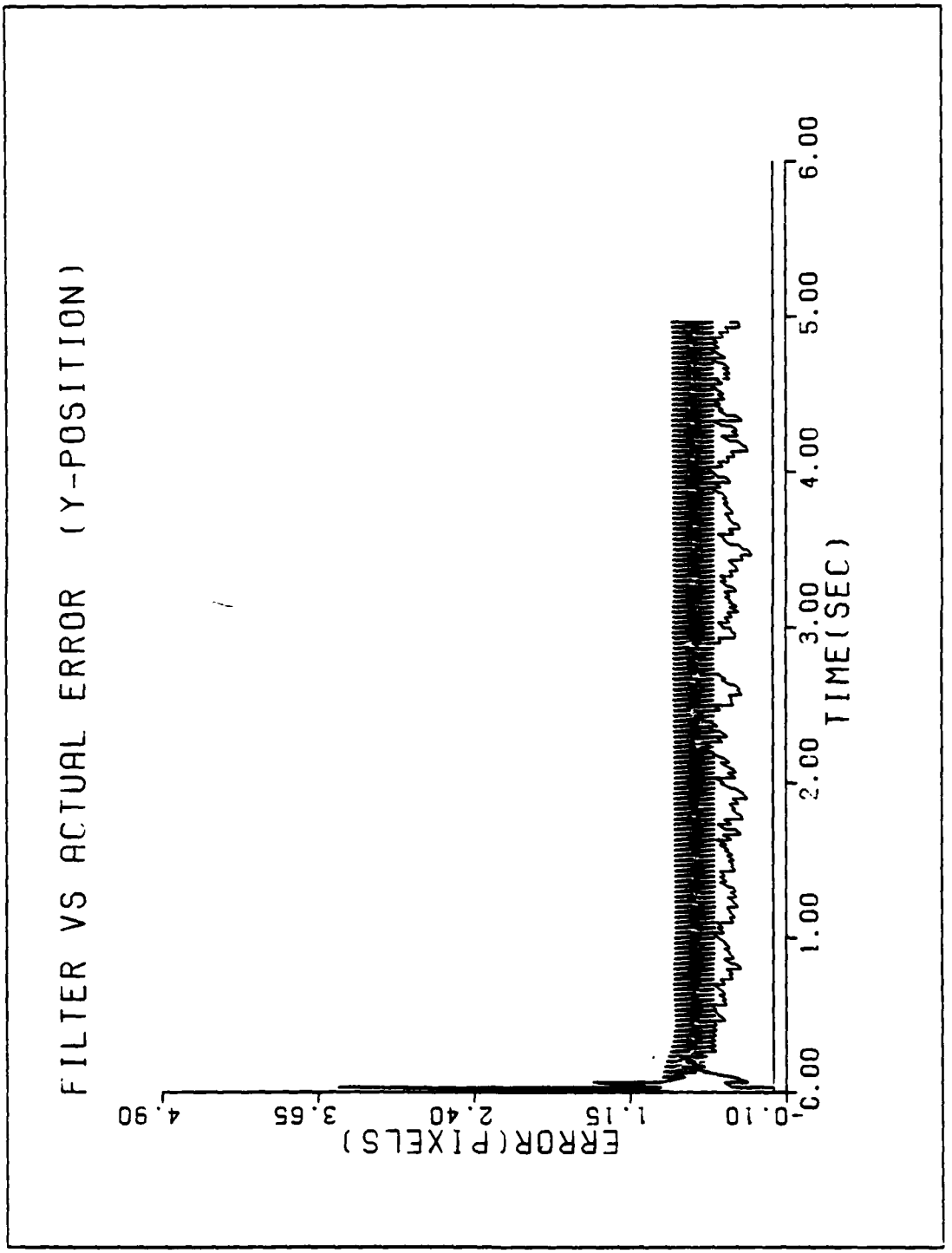


Figure A-3b. Performance Plot for T2GRIM-BASE

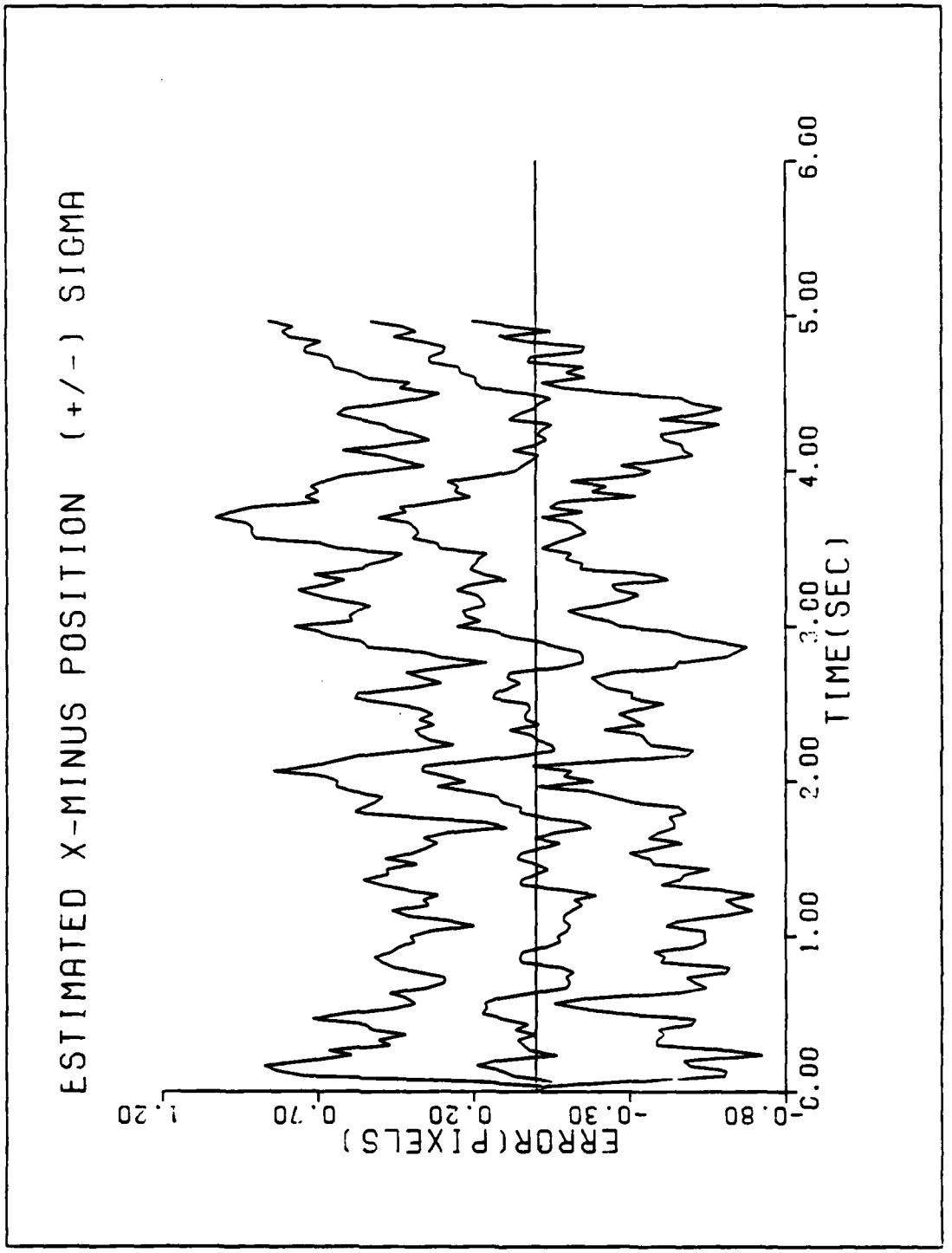


Figure A-3c. Performance Plot for T2G2M1-BASE

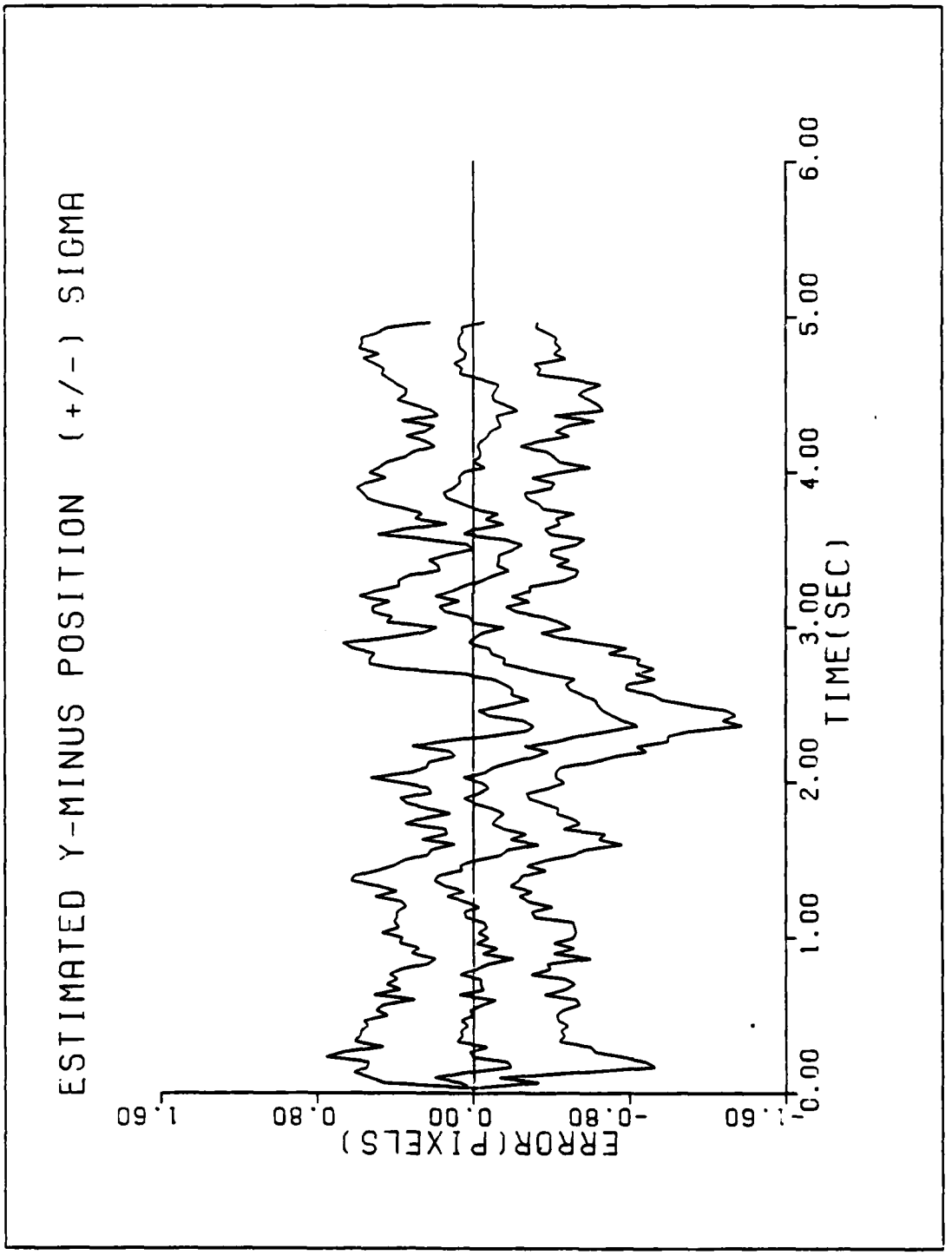


Figure A-3d. Performance Plot for T2G2MM-BASE

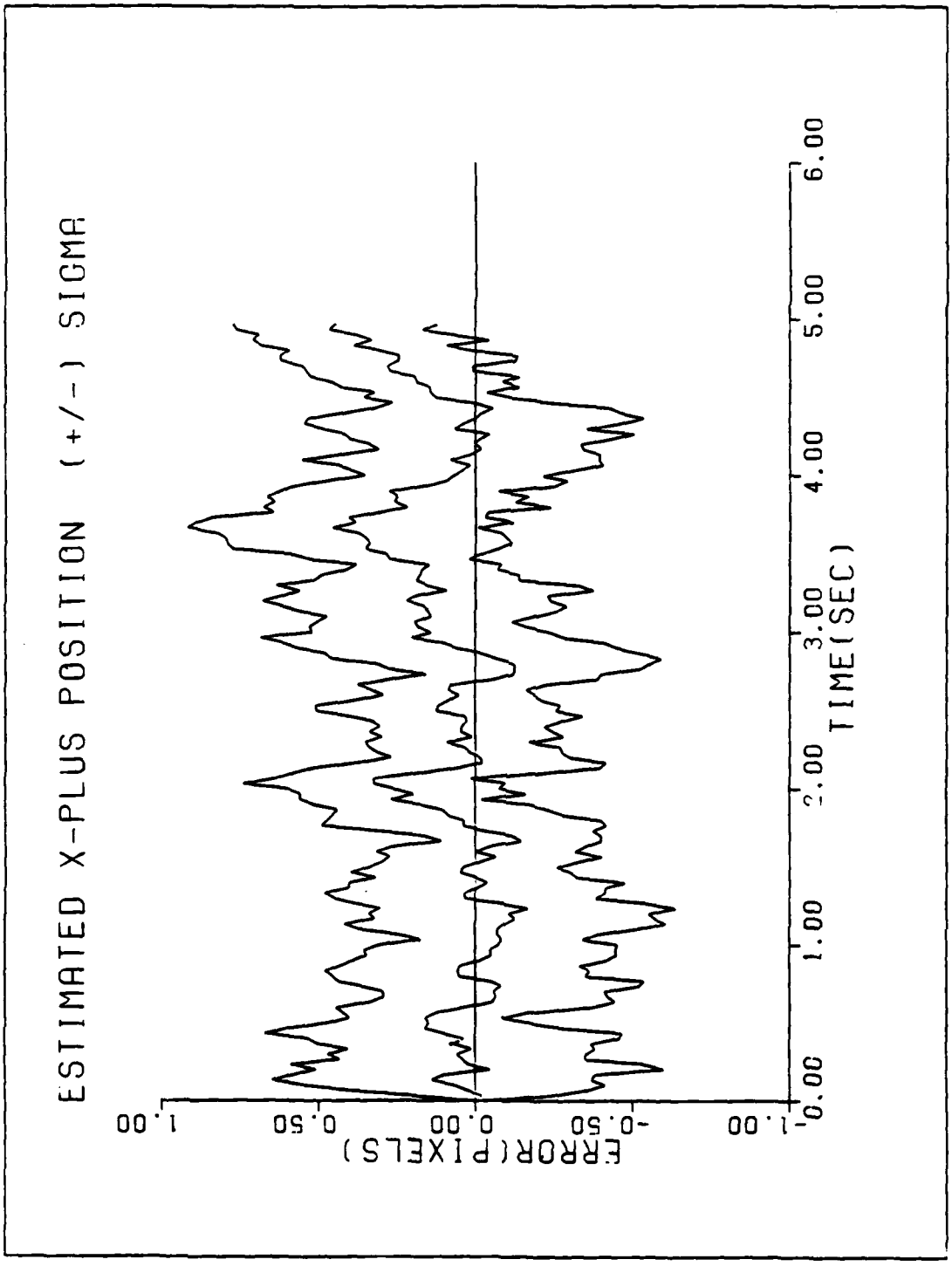


Figure A-3e. Performance Plot for T2G2M1-BASE

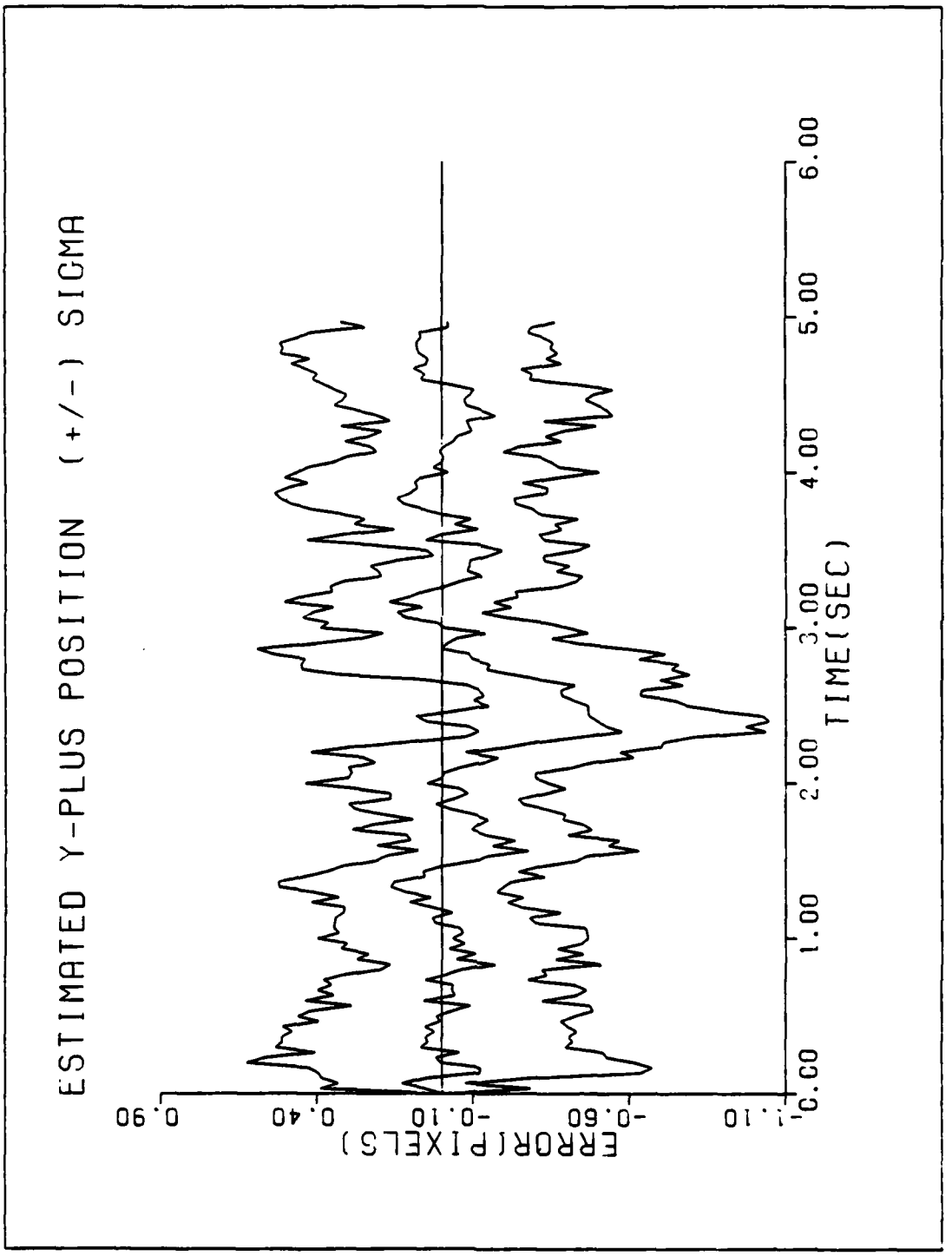


Figure A-3f. Performance Plot for T2G2TM-BASE

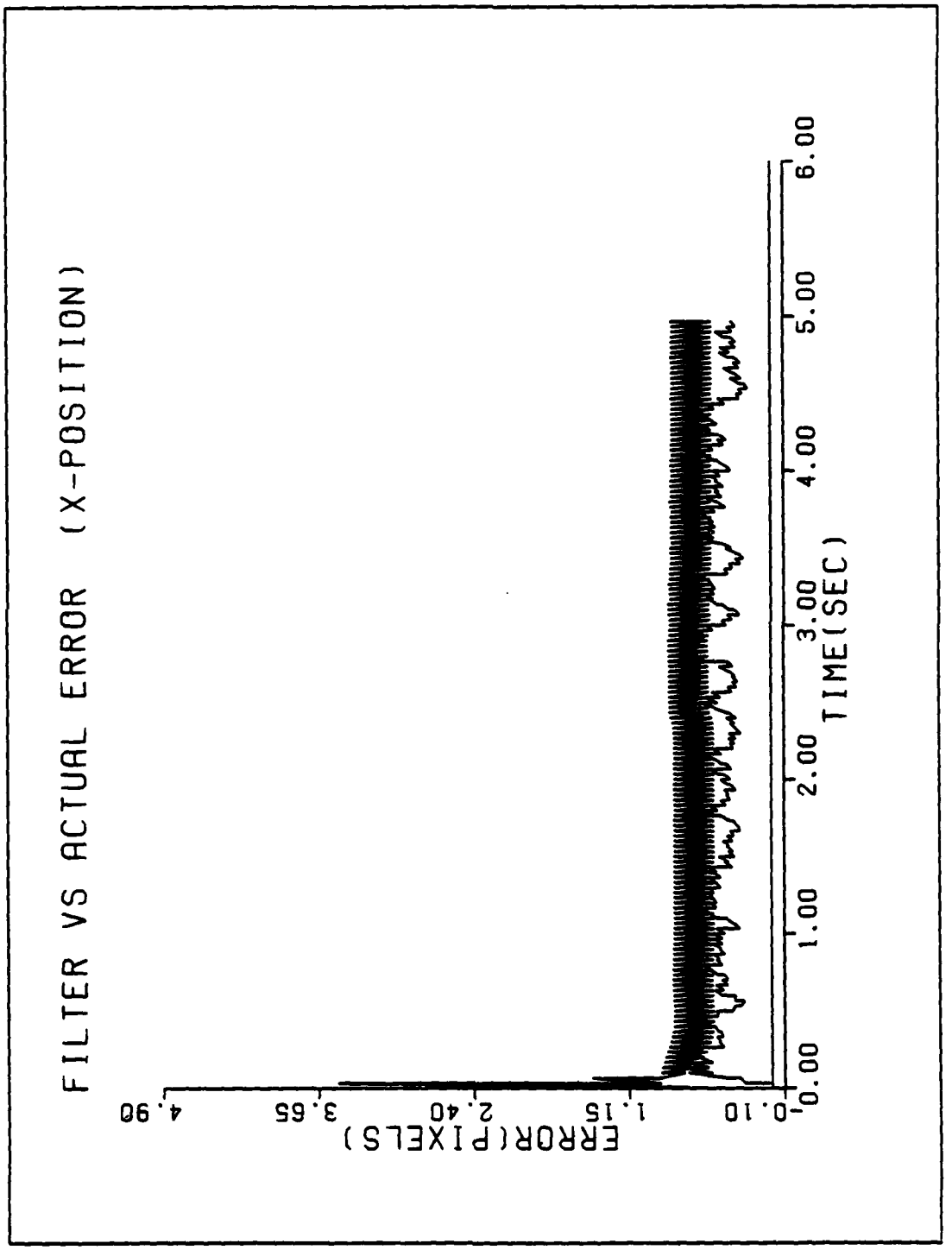


Figure A-4a. Performance Plot for T2G10MM-BASE

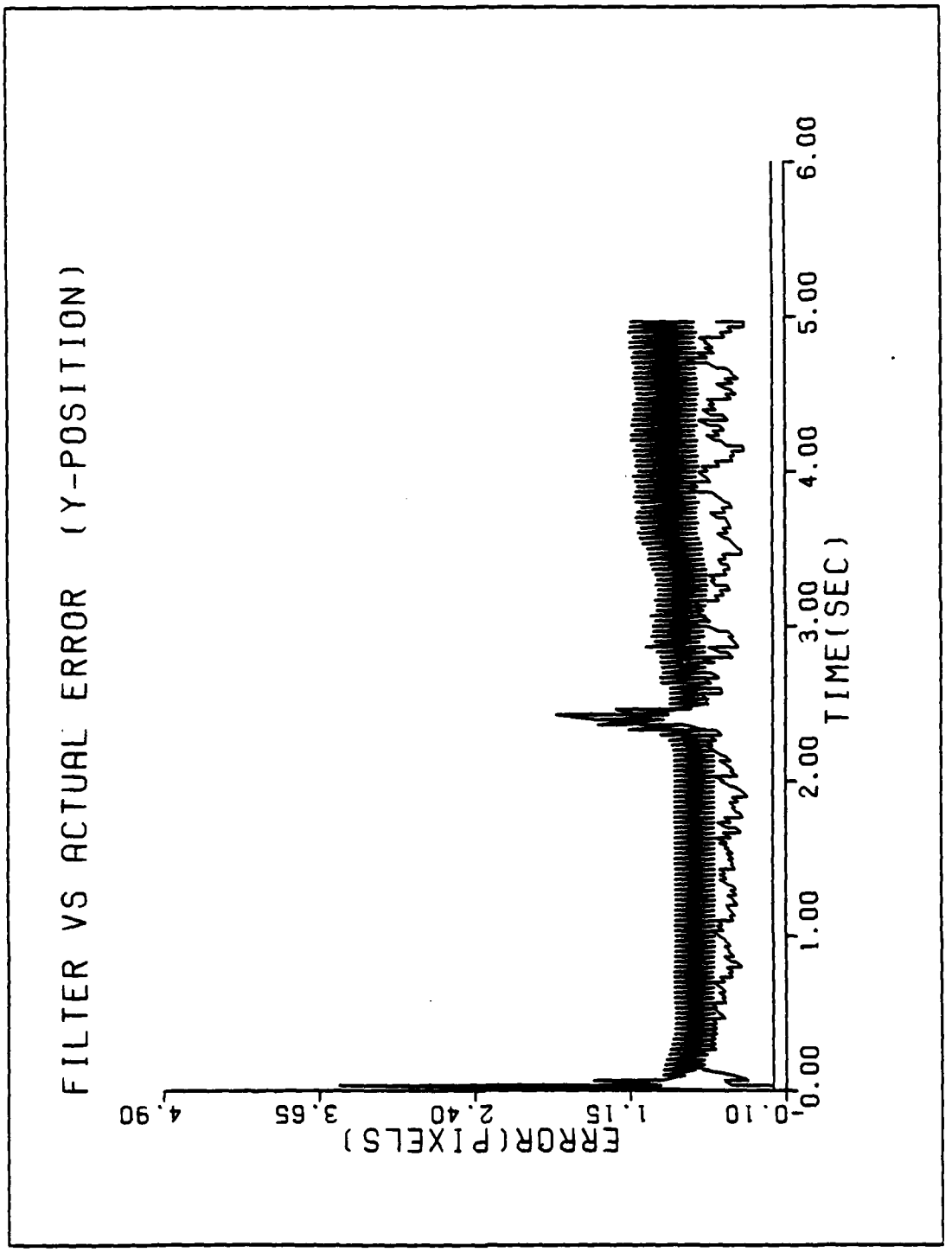


Figure A-4b. Performance Plot for T2G10:D1-BASE

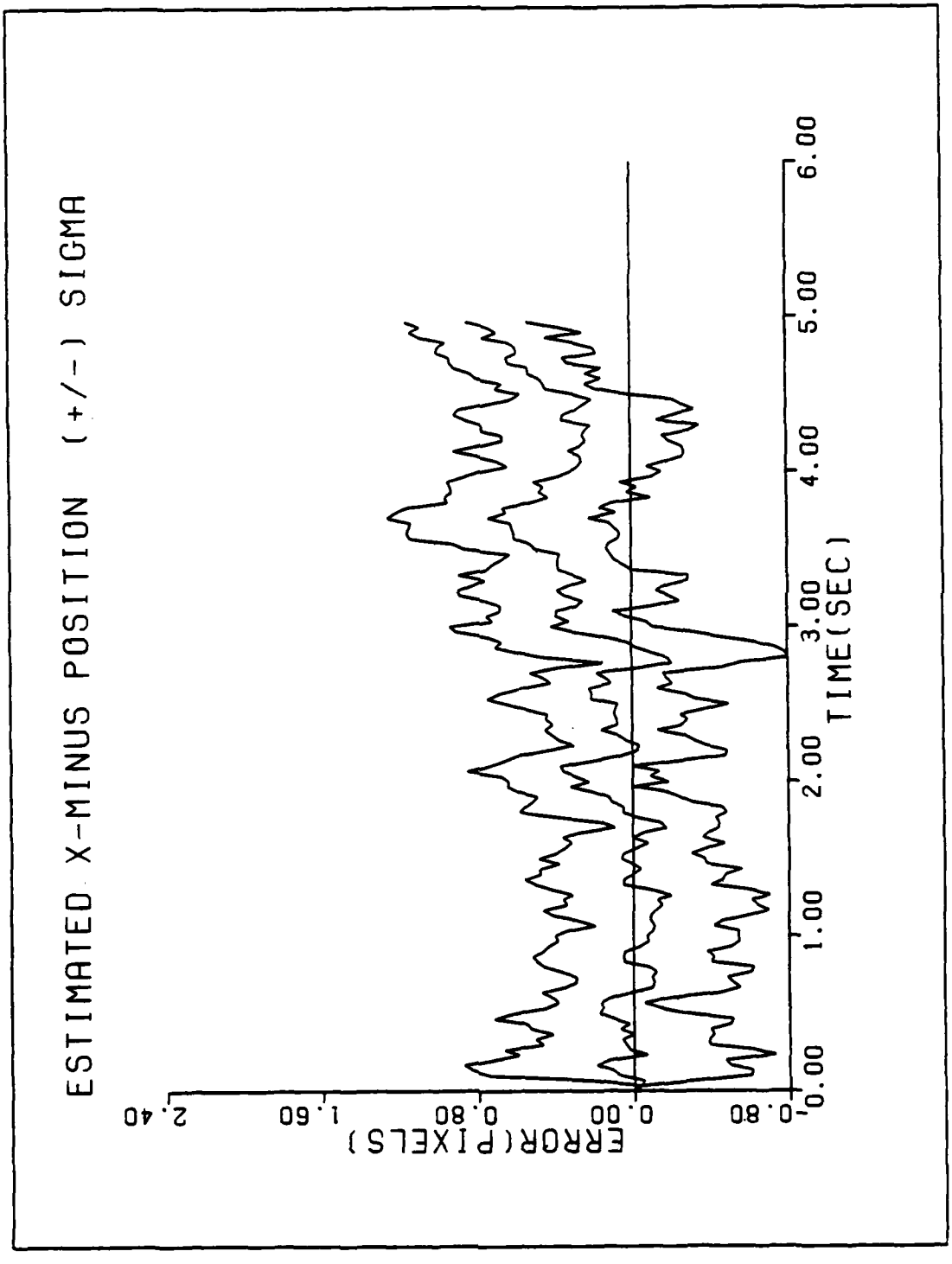


Figure A-4c. Performance Plot for T2G10MM-BASE

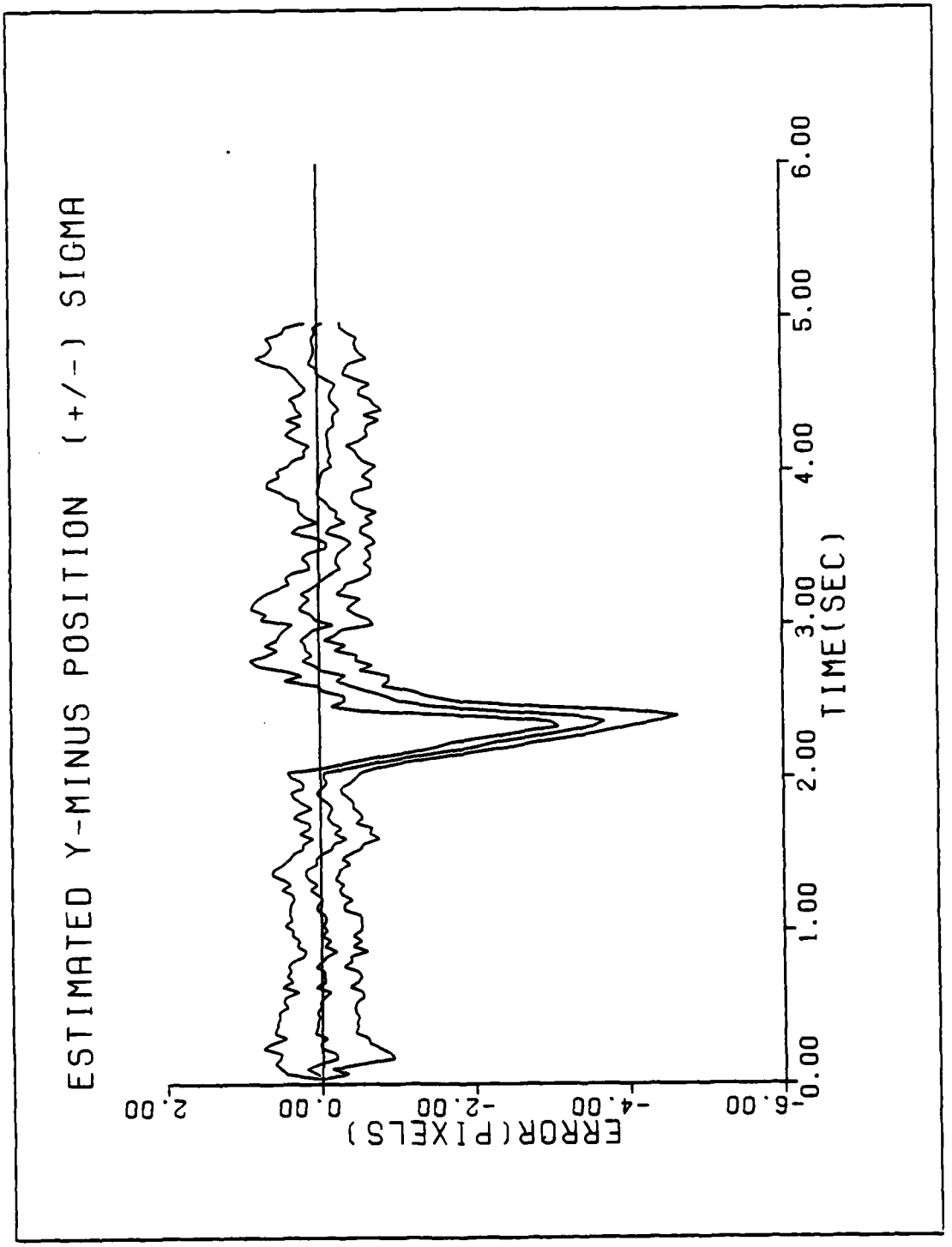


Figure A-4d. Performance Plot for T2G10M1-BASE

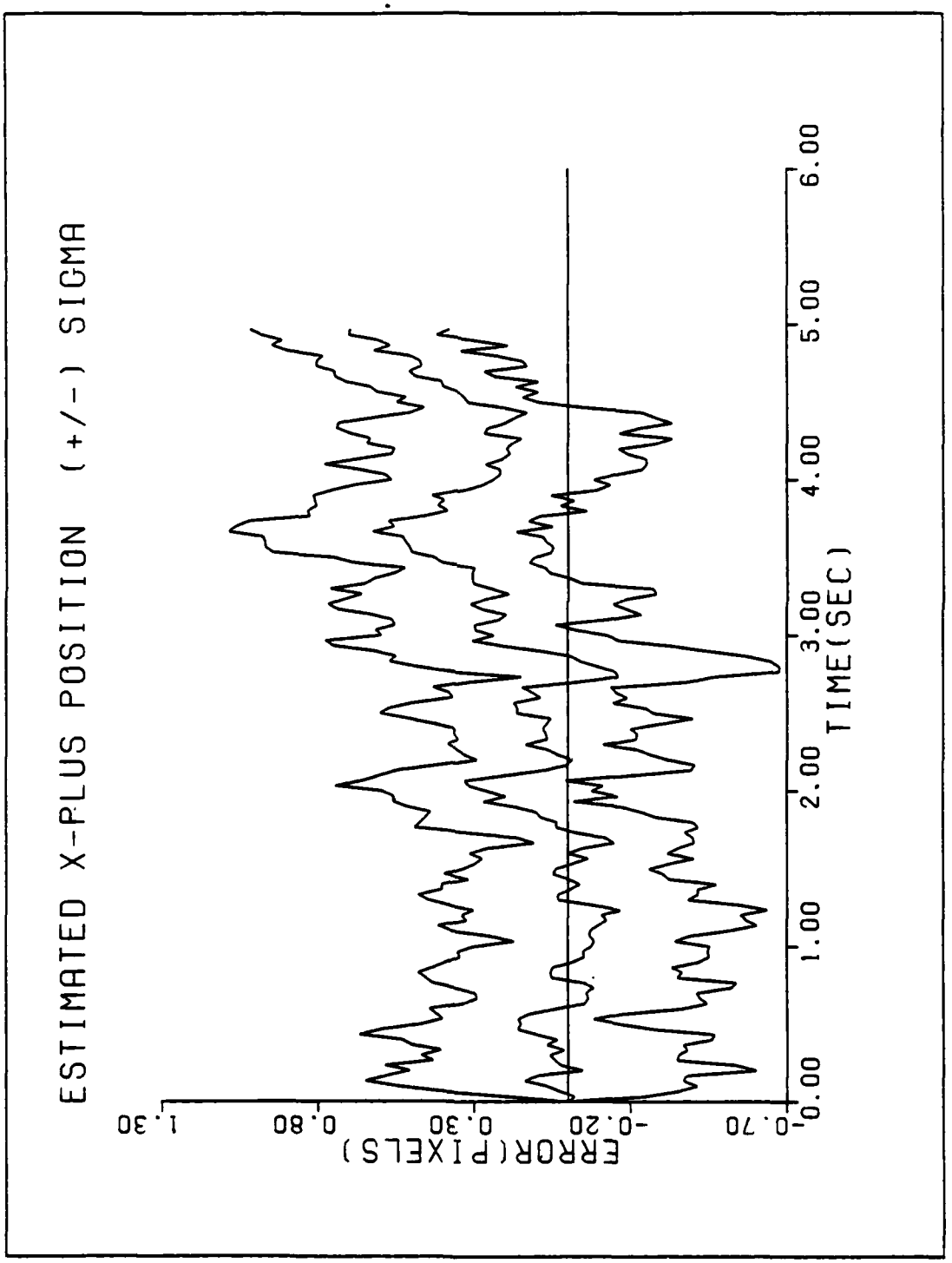


Figure A-5d. Performance Plot for T2G10MM-BASE

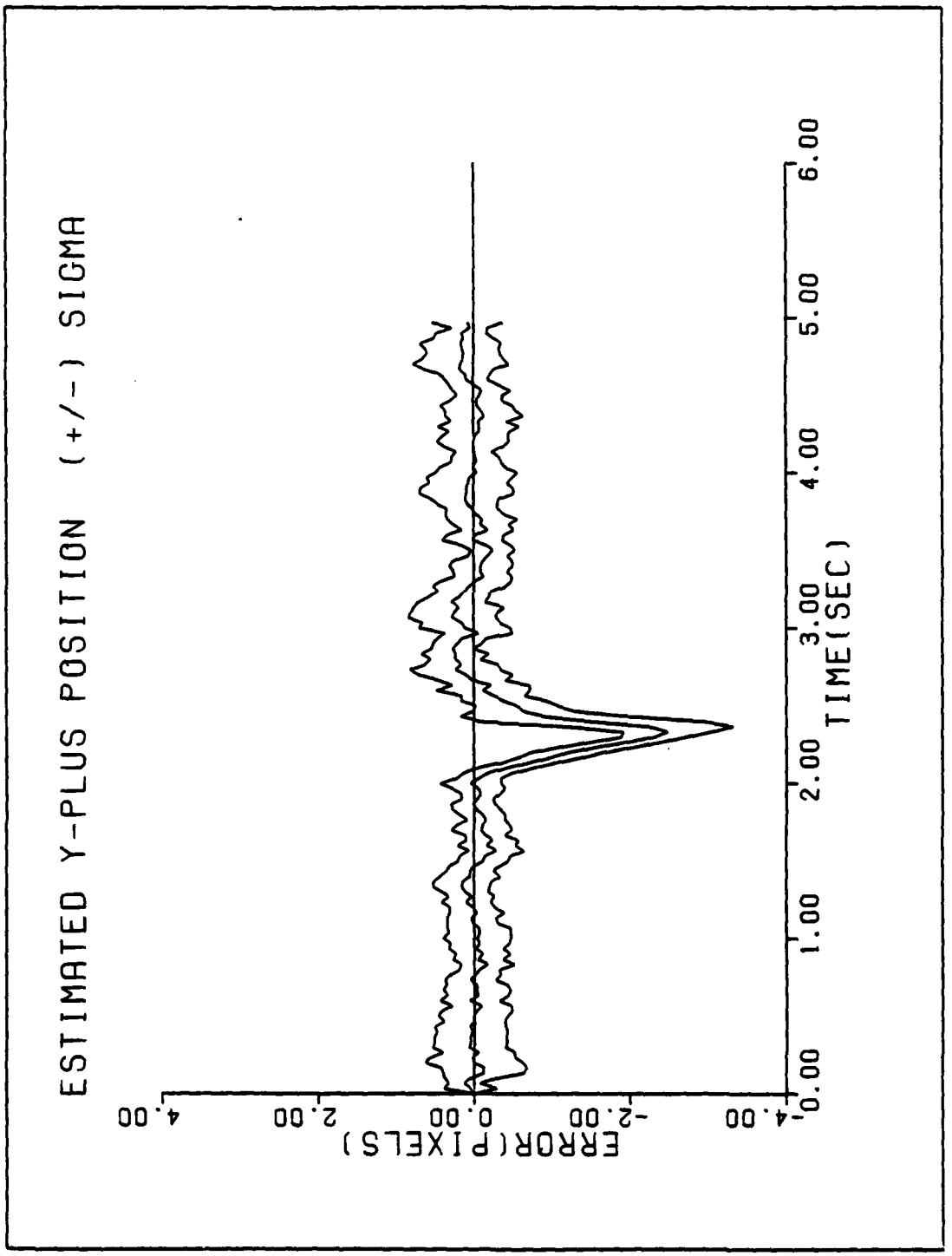


Figure A-5e. Performance Plot for T2G10M1-BASE

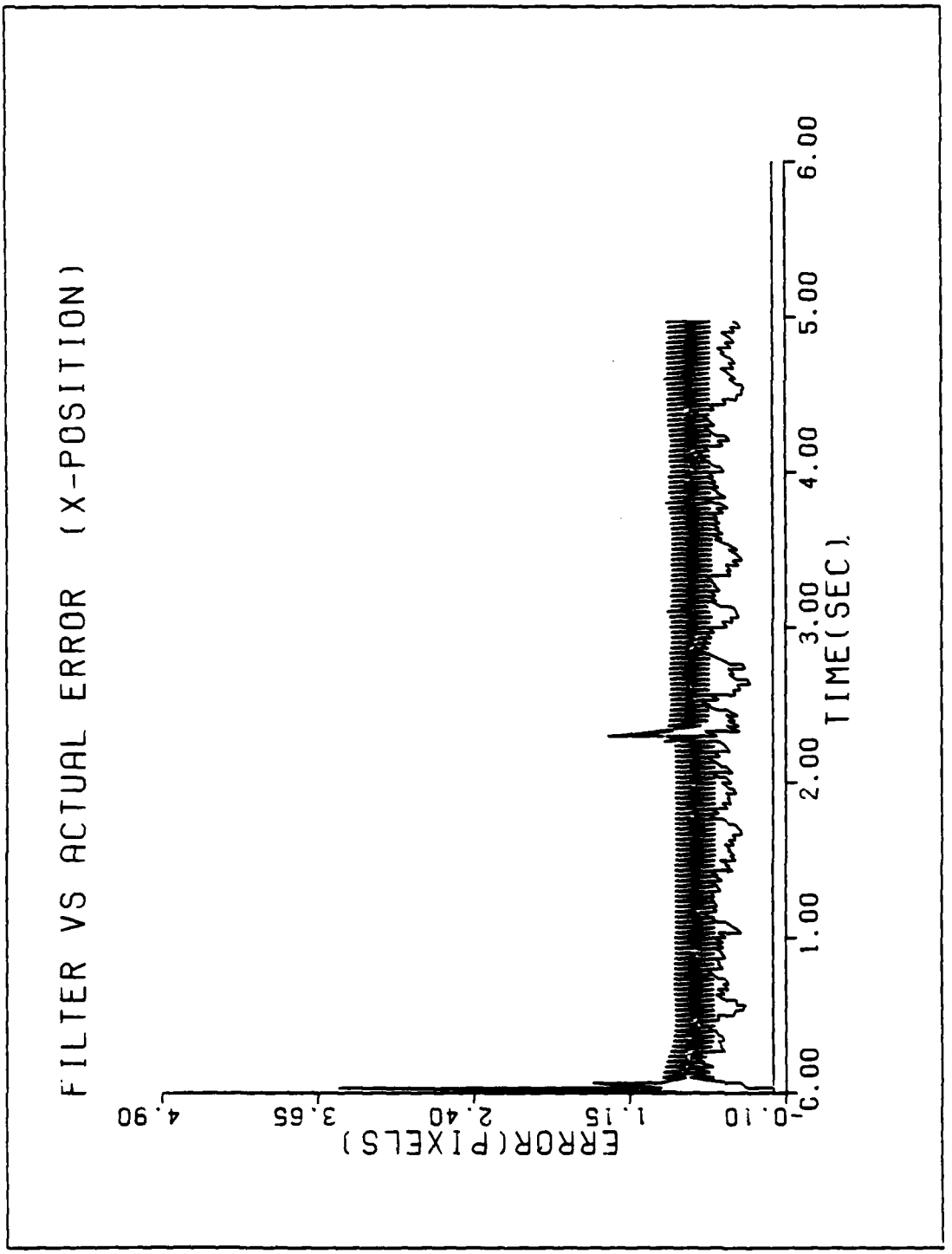


Figure A-6a. Performance Plot for T2G2OMI-BASE

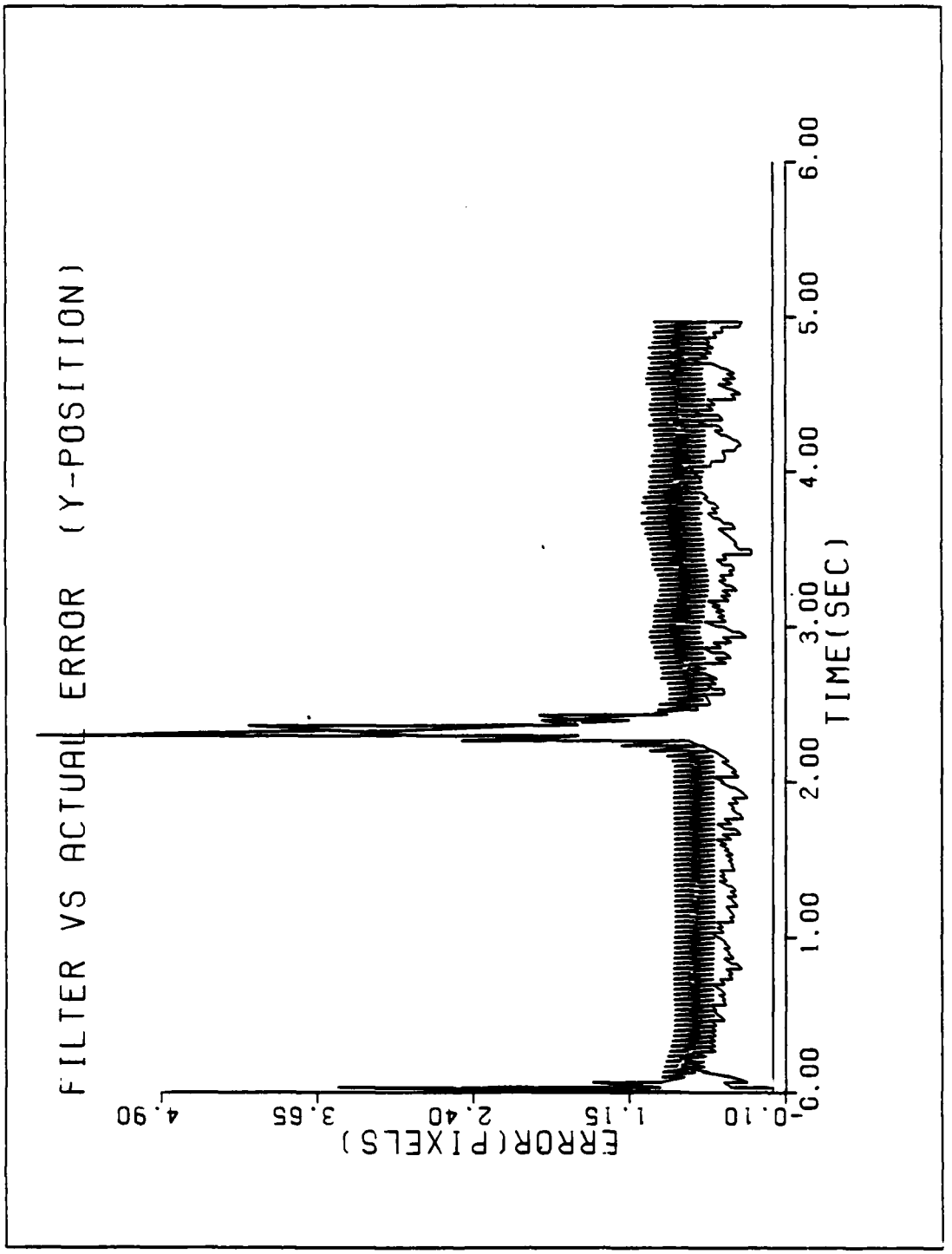


Figure A-6b. Performance Plot for T2G20'T-BASE

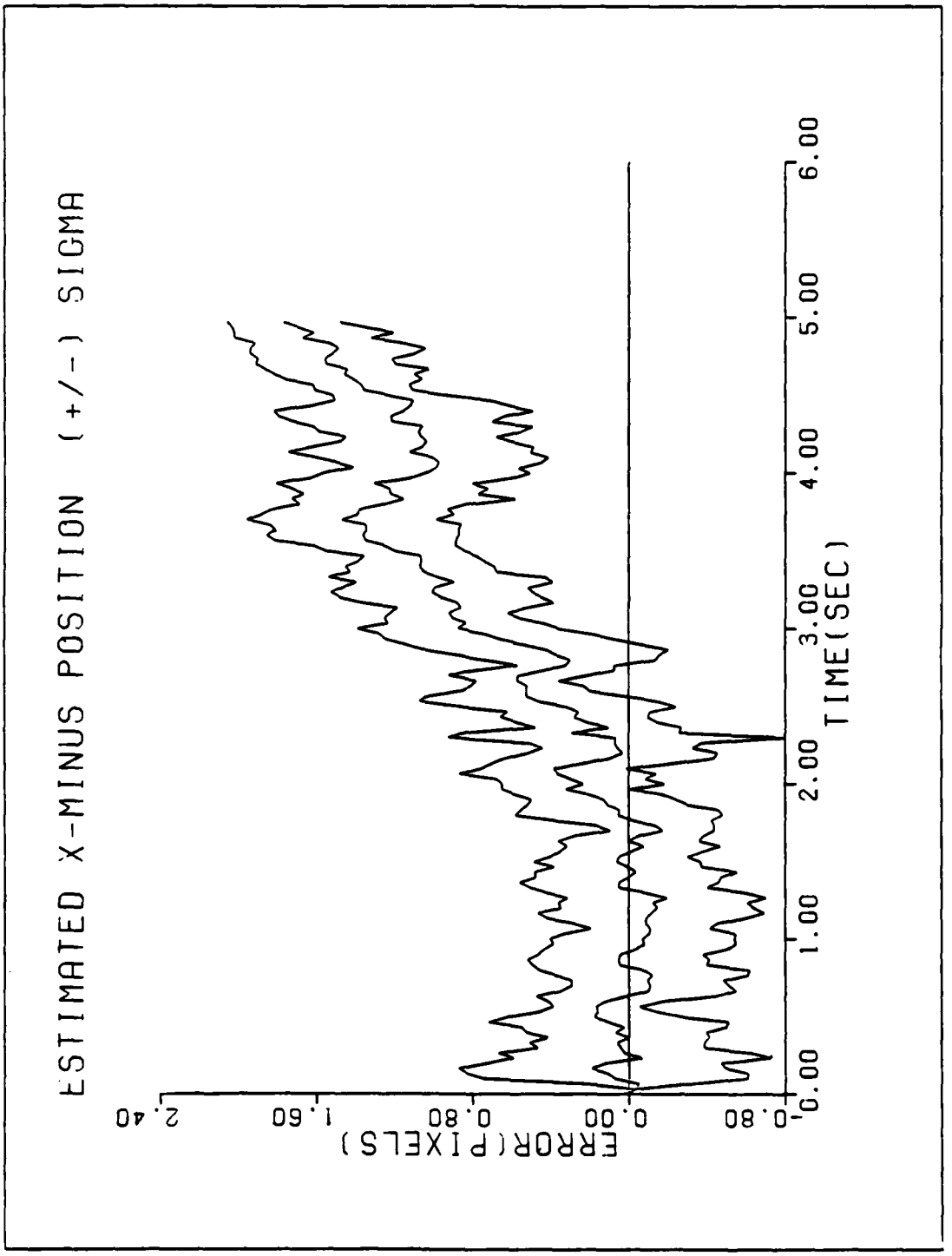


Figure A-6c. Performance Plot for T2G20FF-BASE

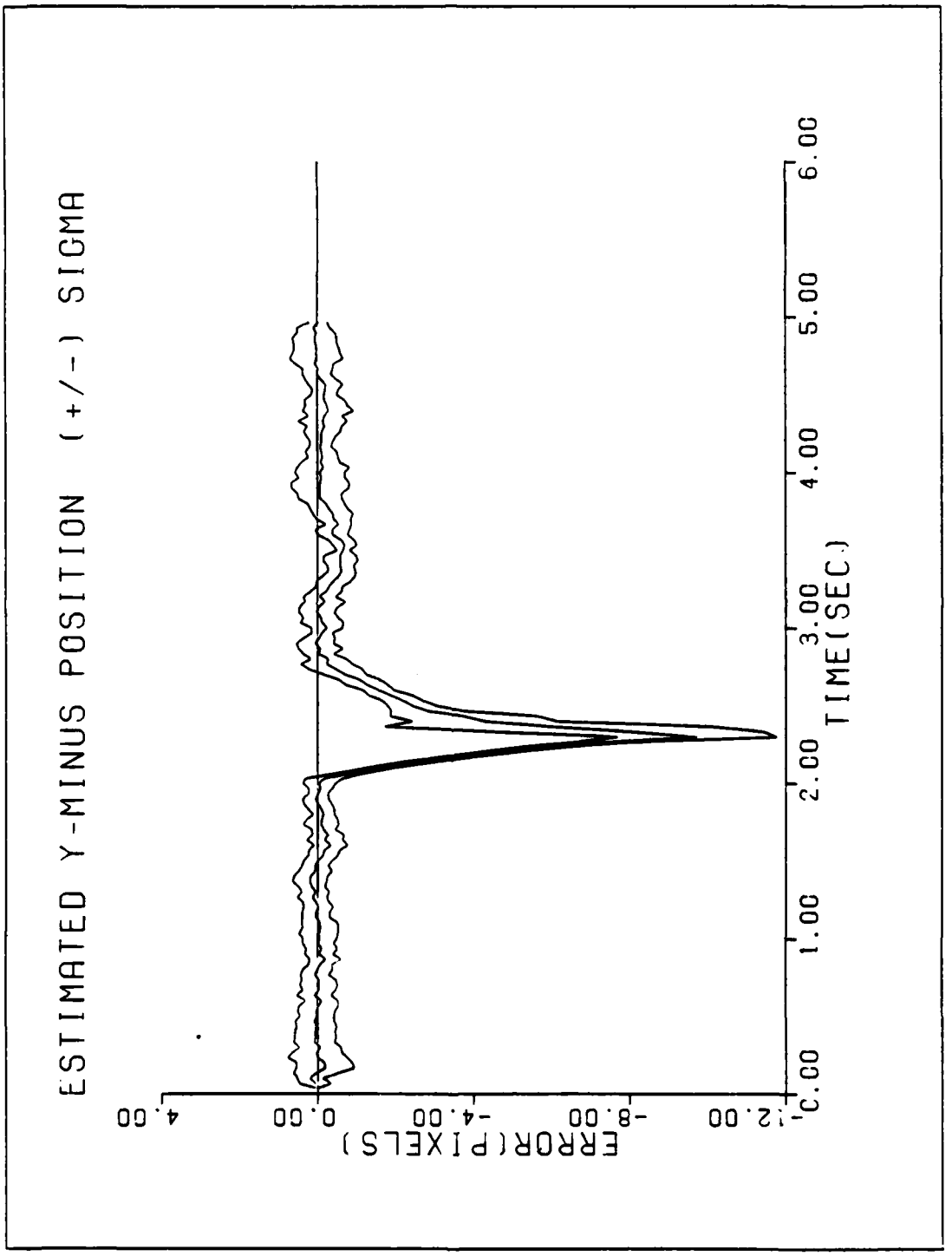


Figure A-6d. Performance Plot for T2G2DIFI-BASE

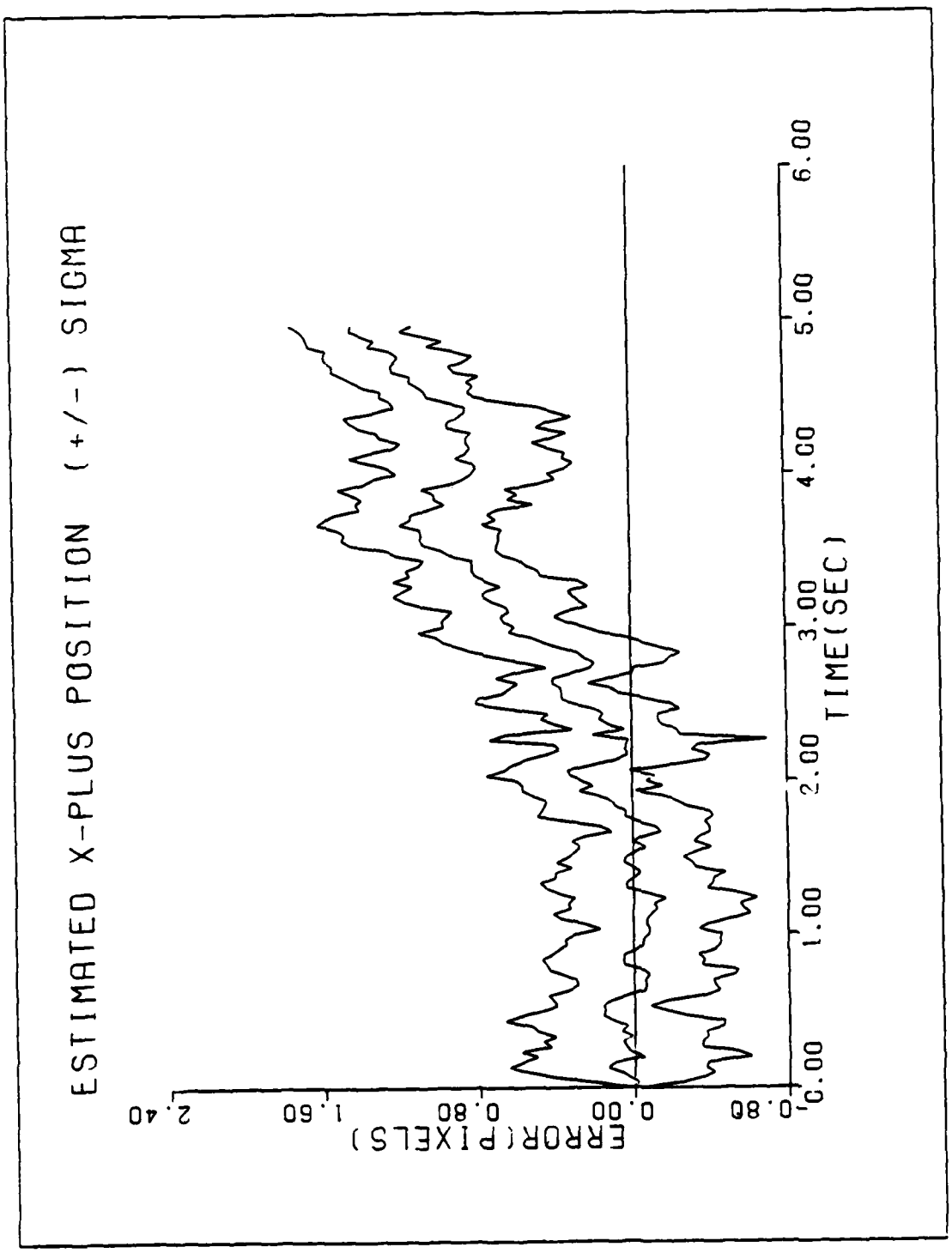


Figure A-6e. Performance Plot for T2020M1-BASE

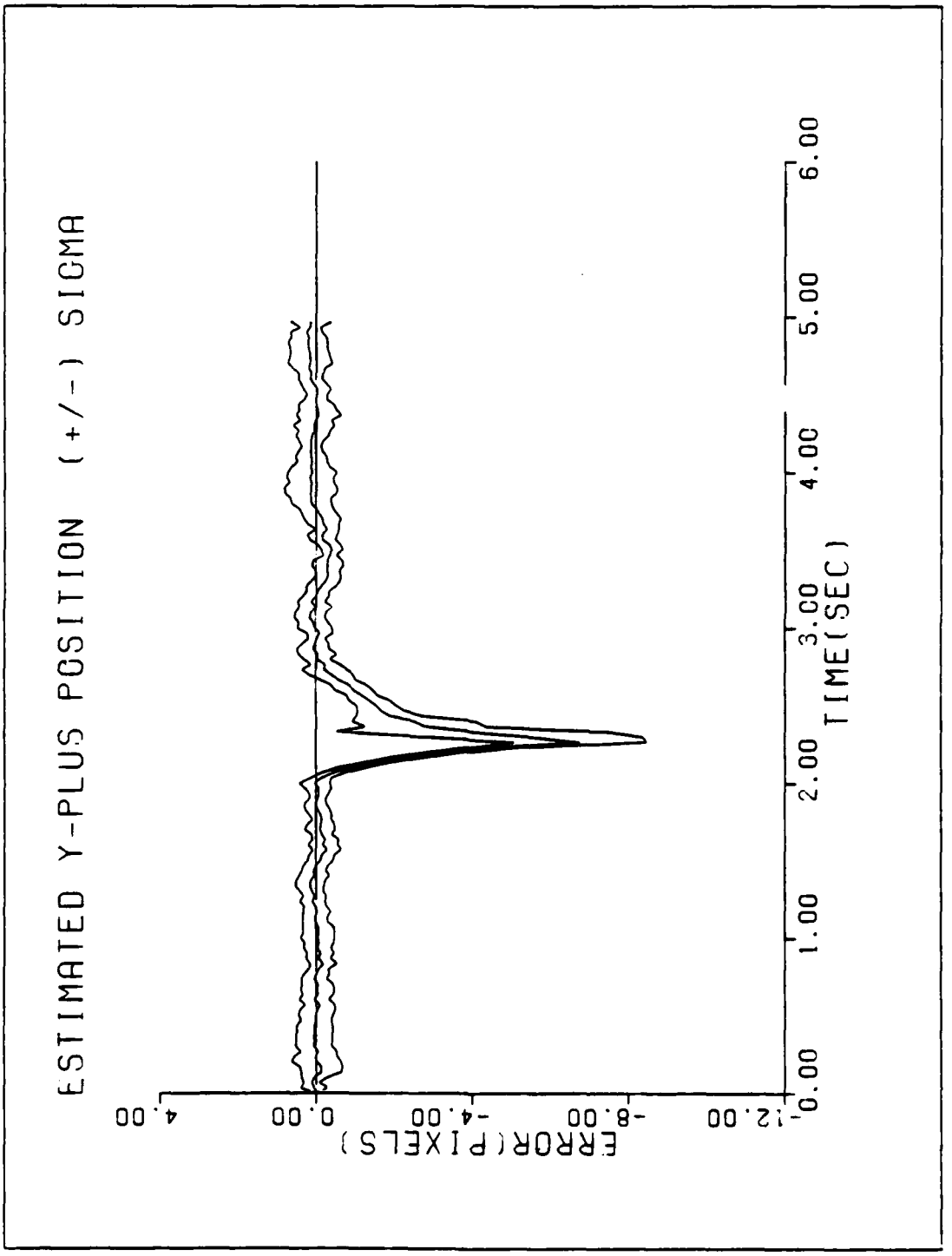


Figure A-6f. Performance Plot for T2G20IDI-BASE

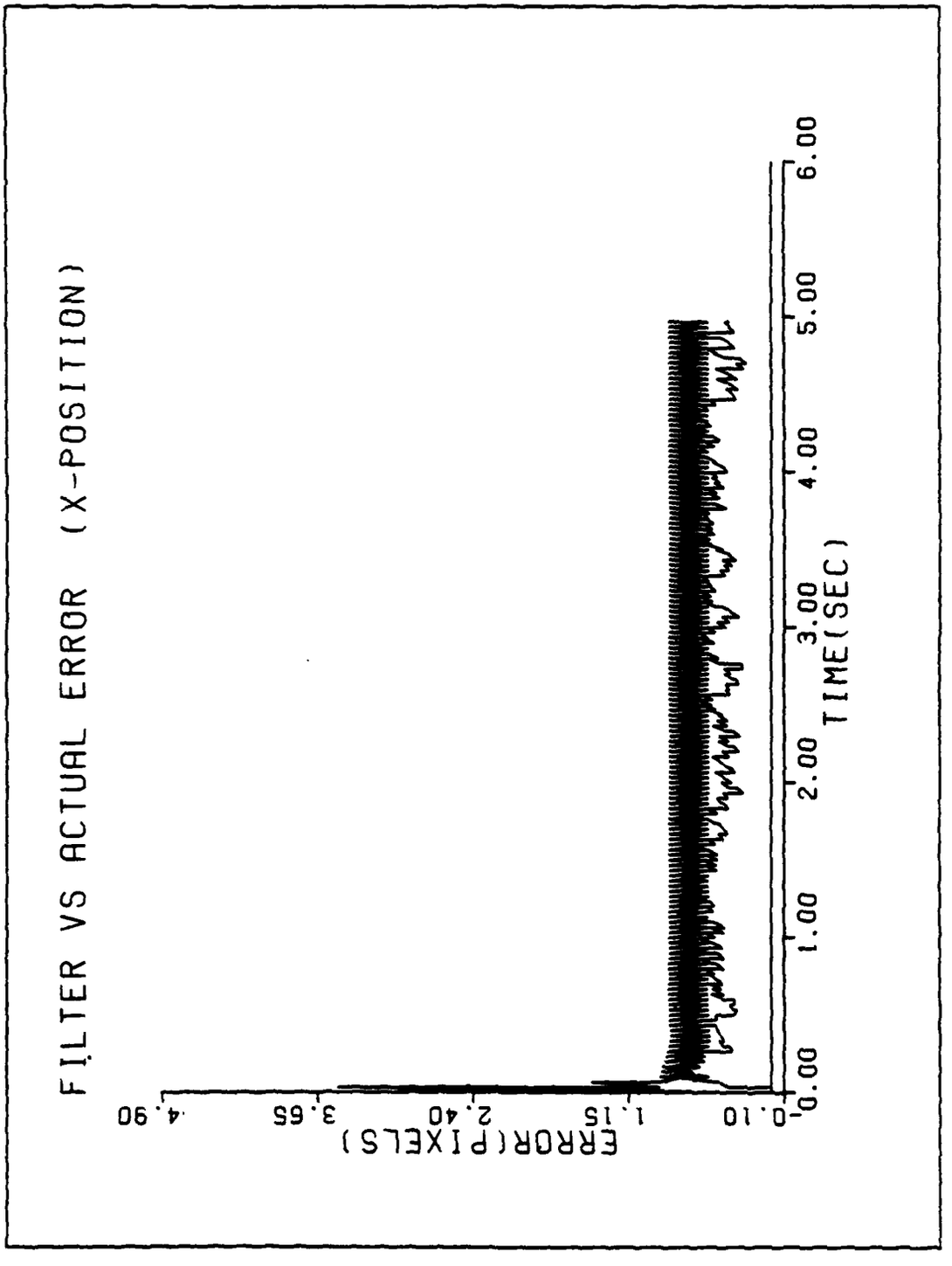


Figure A-7a. Performance Plot for T2020T2-BASE

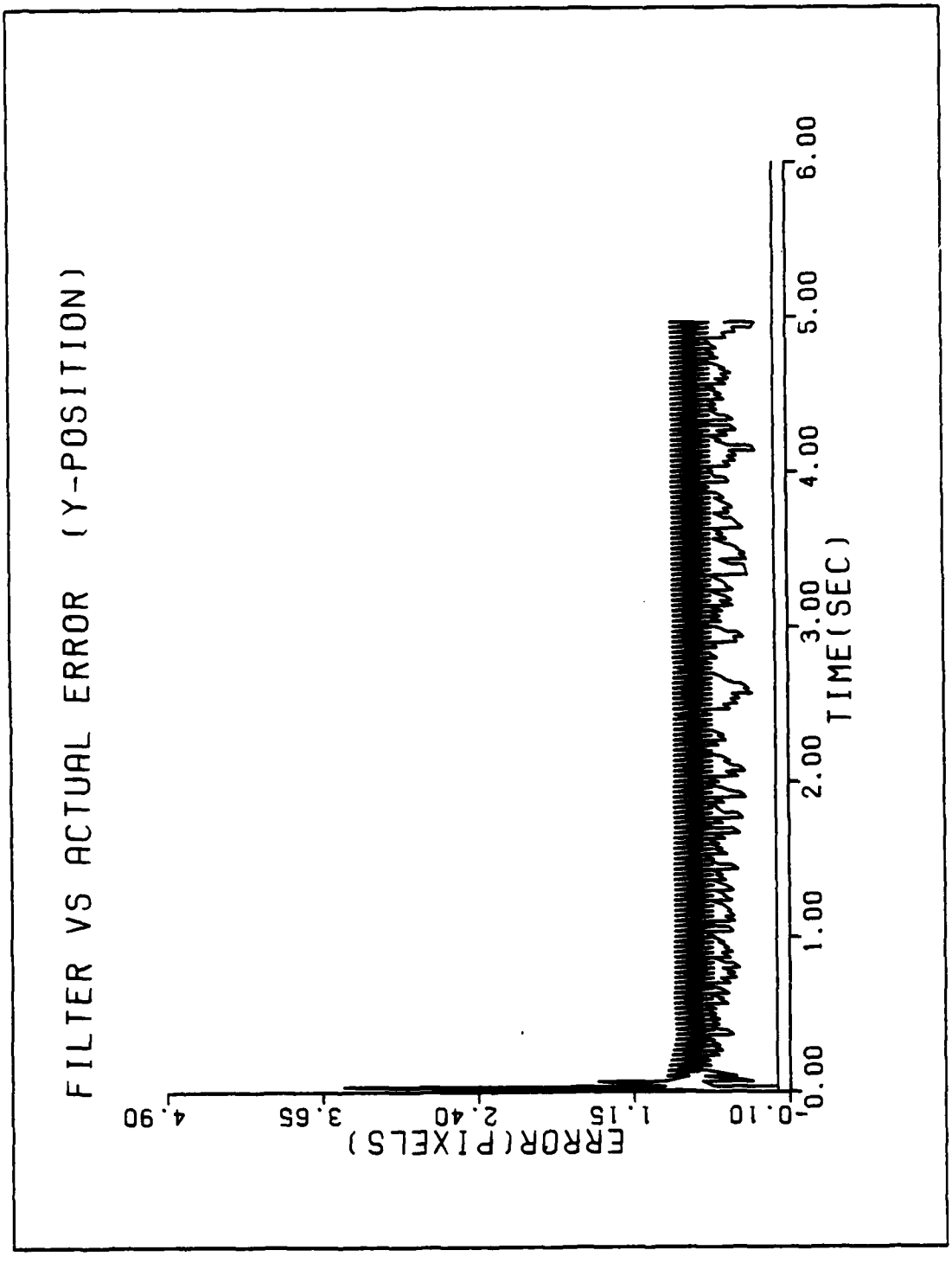


Figure A-7b. Performance Plot for T2G2DT2-BASE

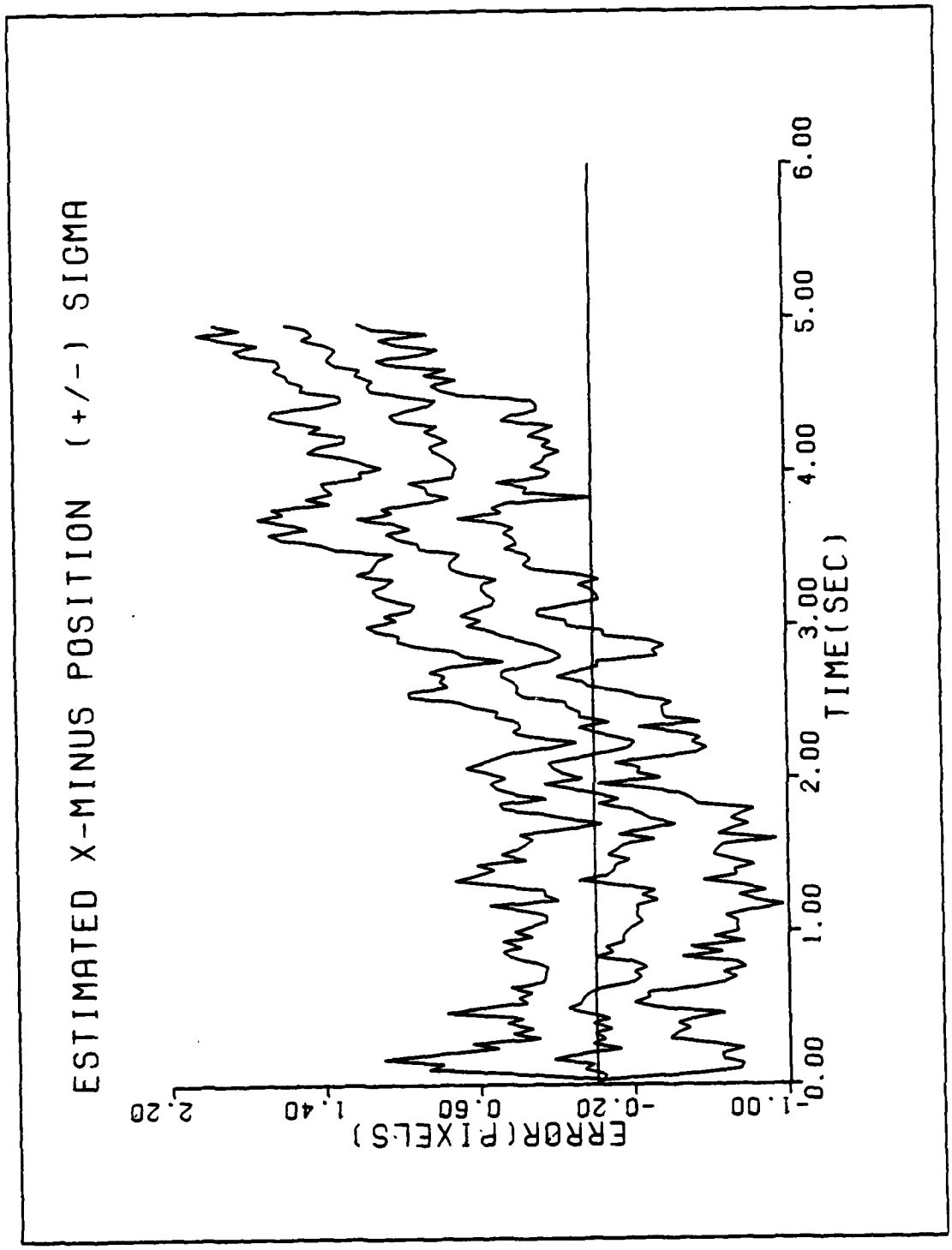


Figure A-7b. Performance Plot for T2GCOT2-BASE

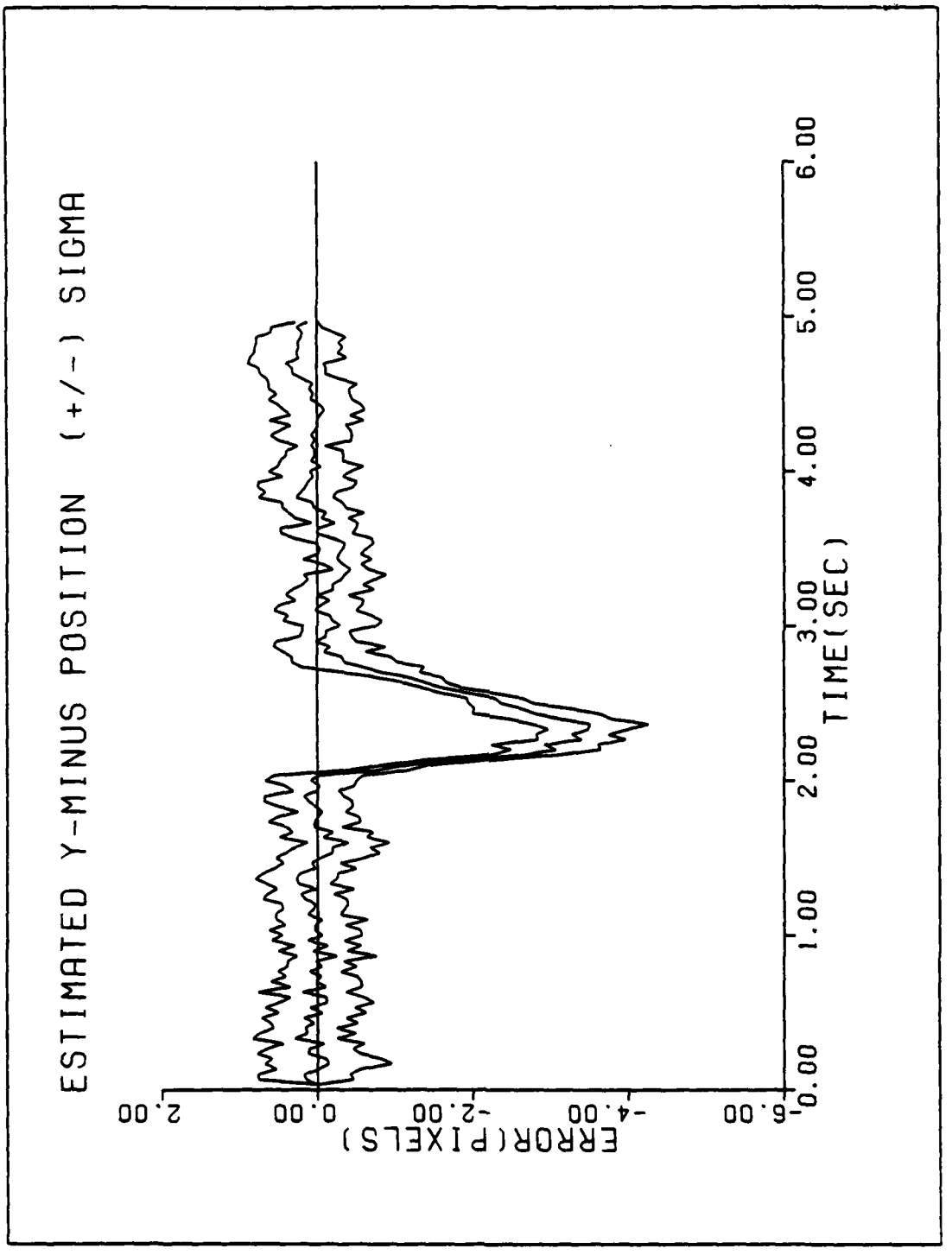


Figure A-7c. Performance Plot for T2602TE-BASE

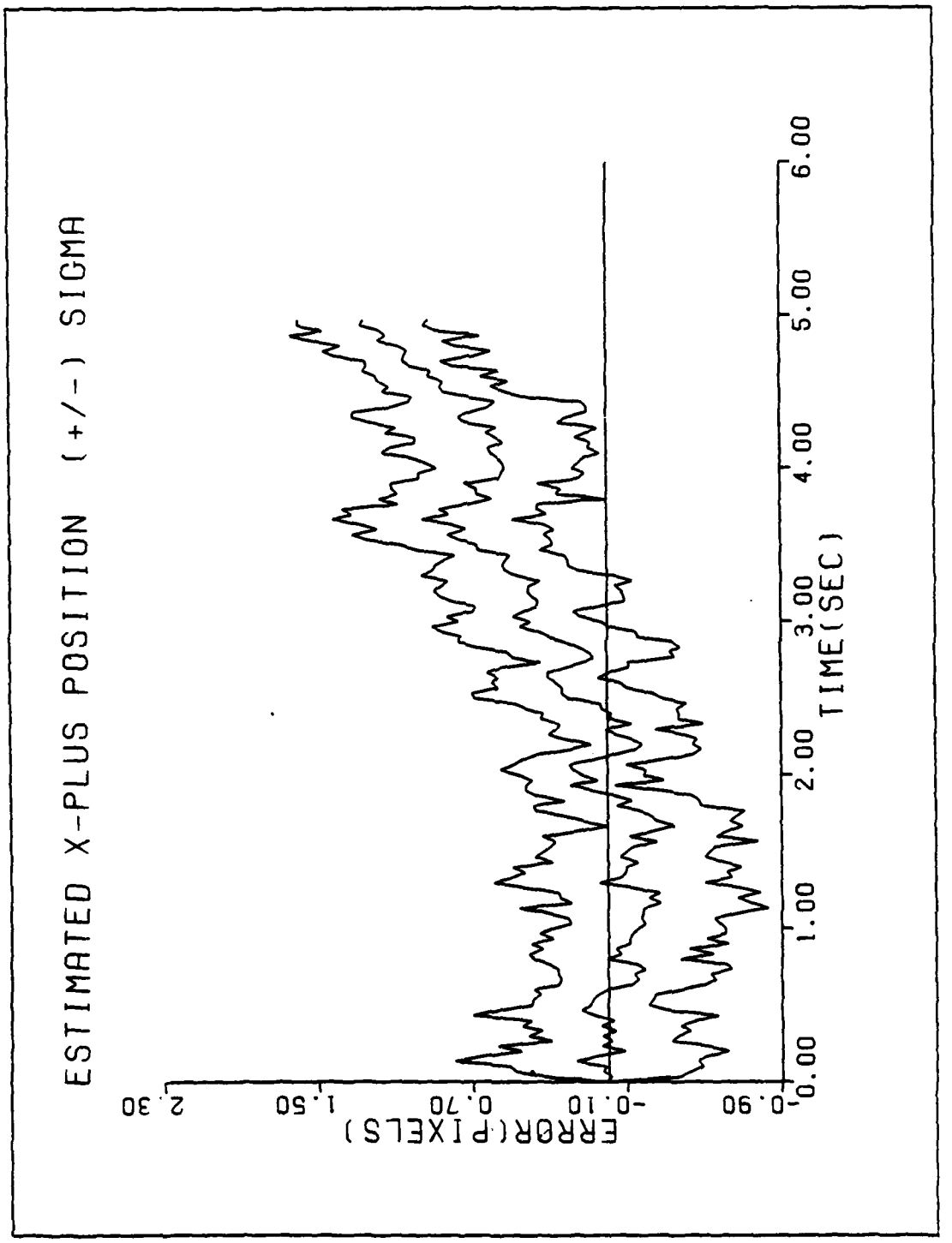


Figure A-7d. Performance Plot for T2G20F2-BASE

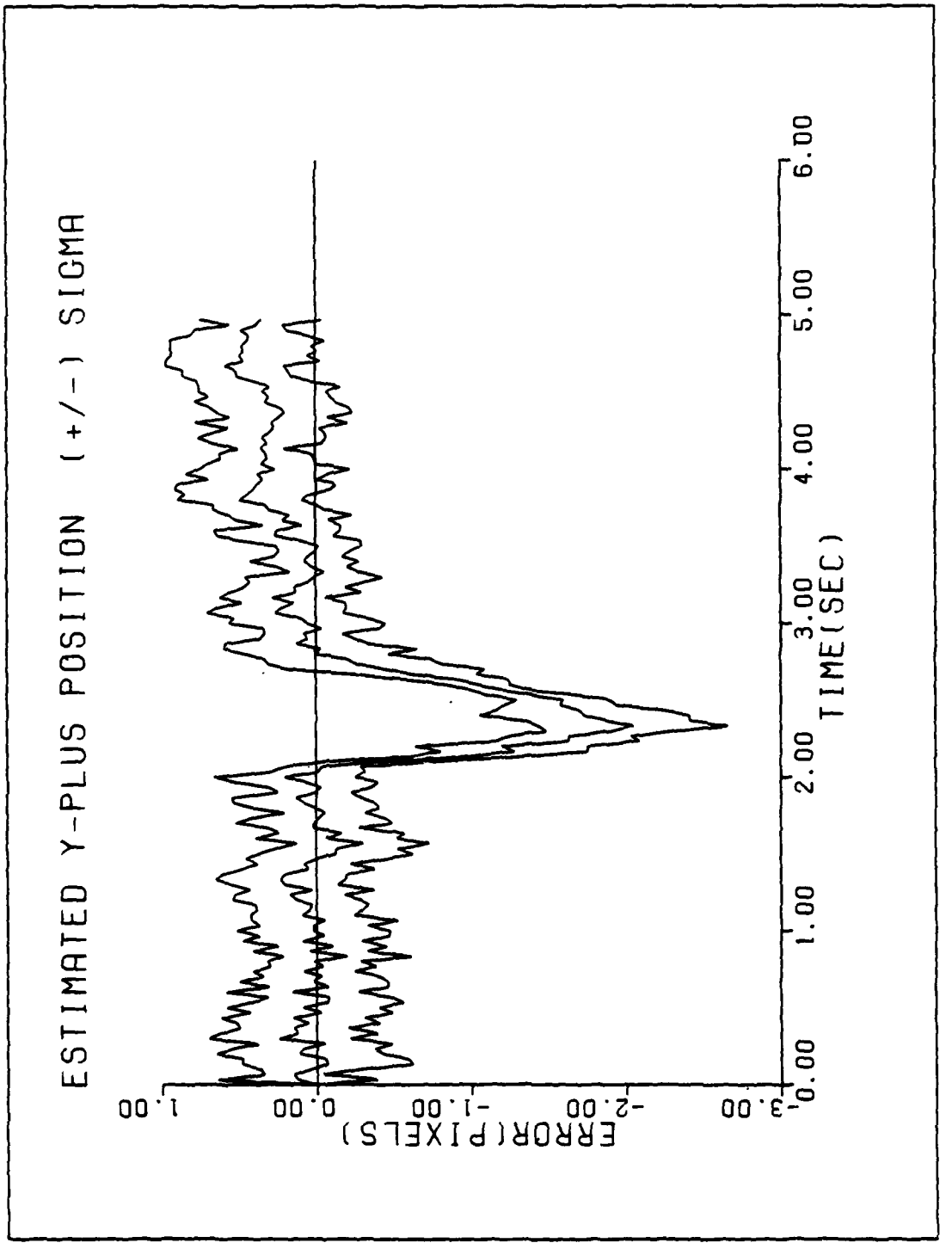


Figure A-7e. Performance Plot for T2G20F2-BASE

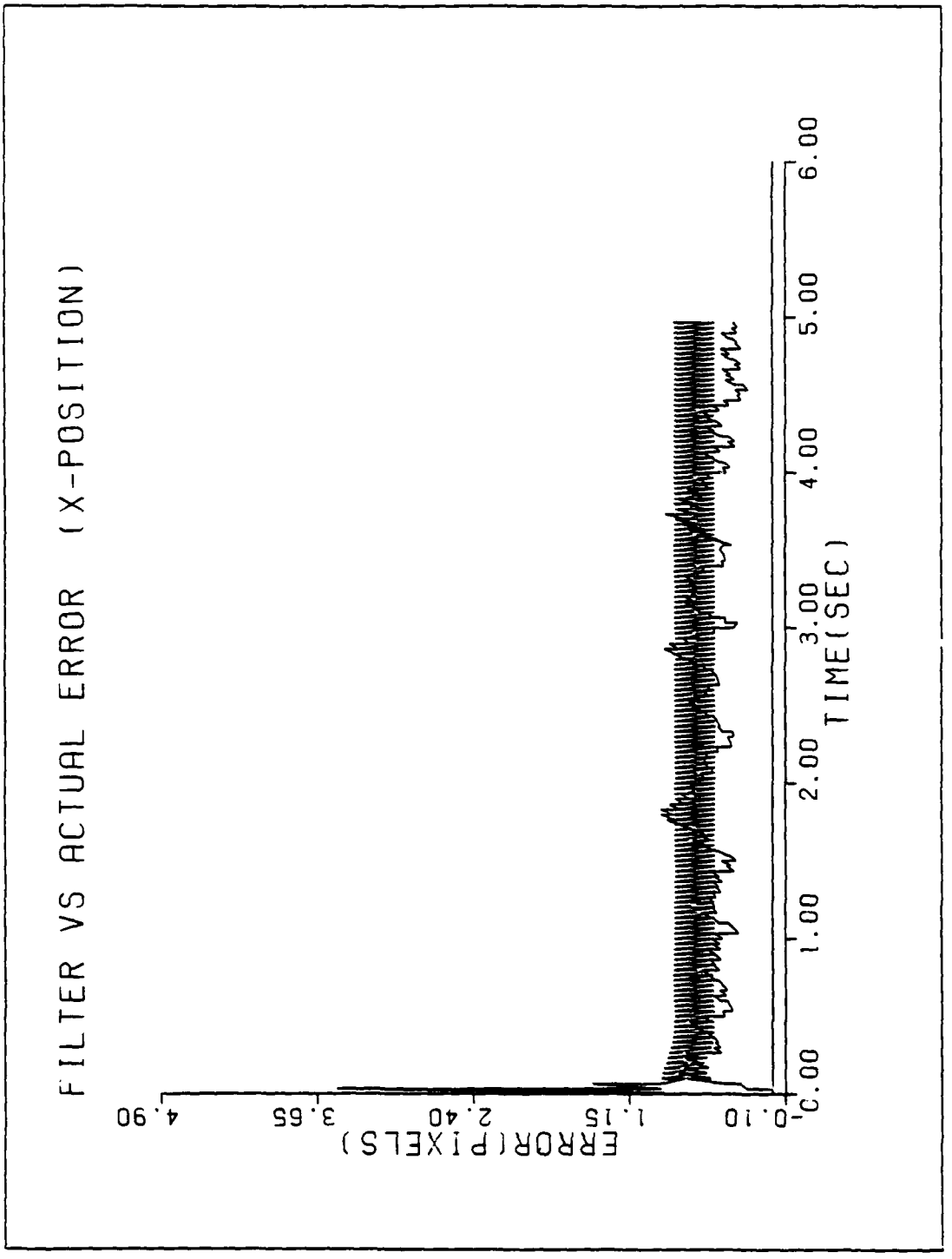


Figure A-8a. Performance Plot for T3G2M-BASE

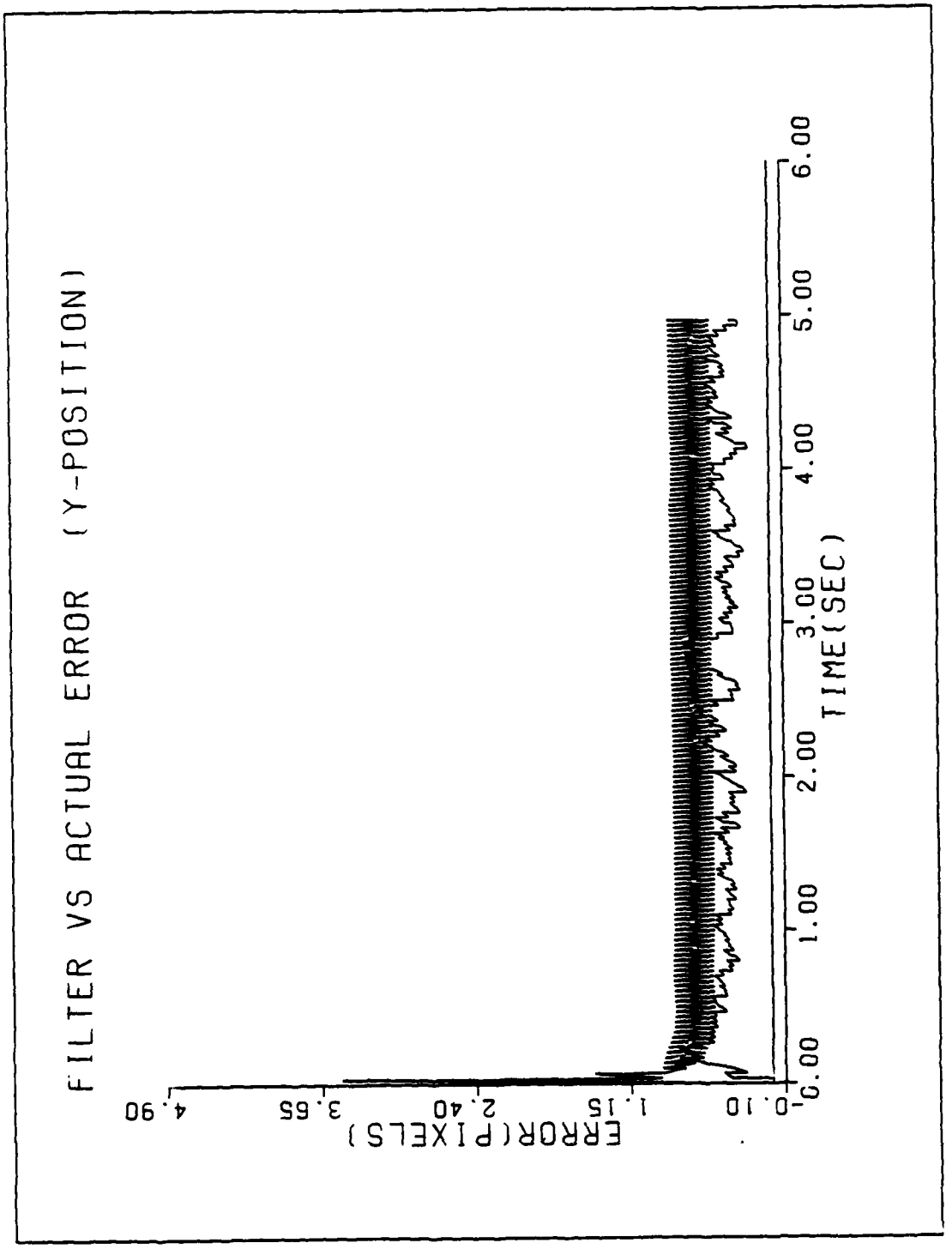


Figure A-8b. Performance Plot for T3G2M-BASE

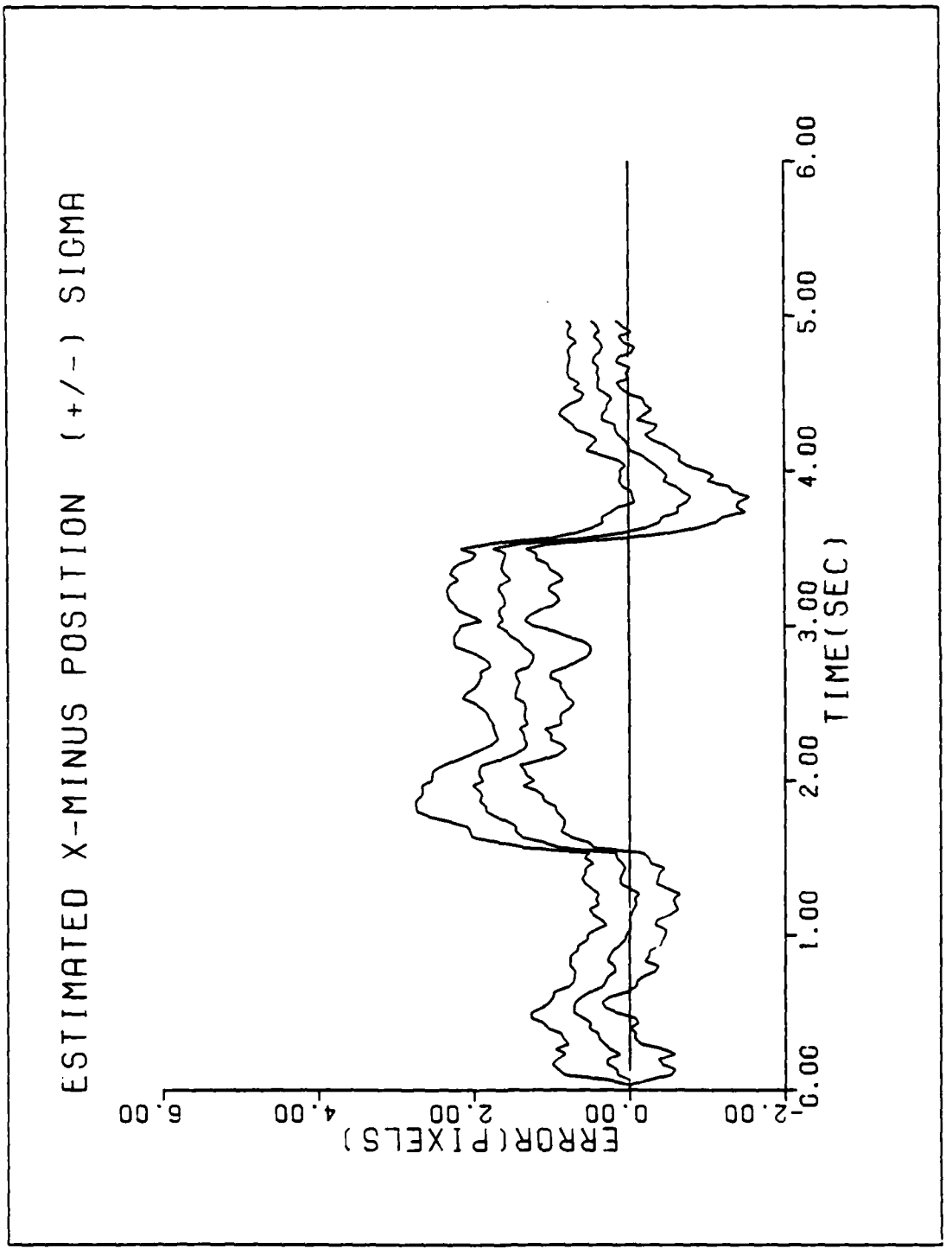


Figure A-8c. Performance Plot for T3G2M-BASE

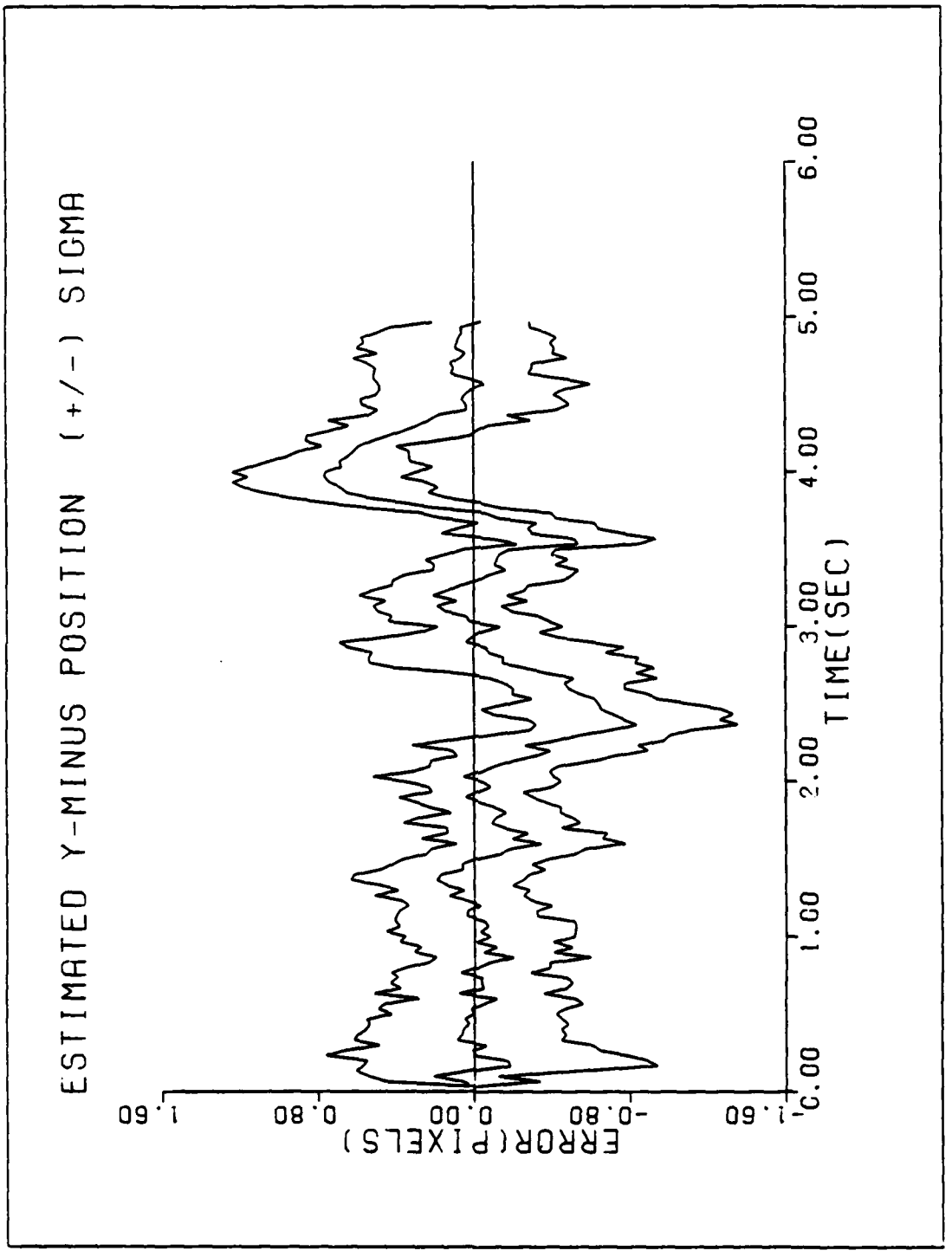


Figure A-8d. Performance Plot for T3G2M-BASE

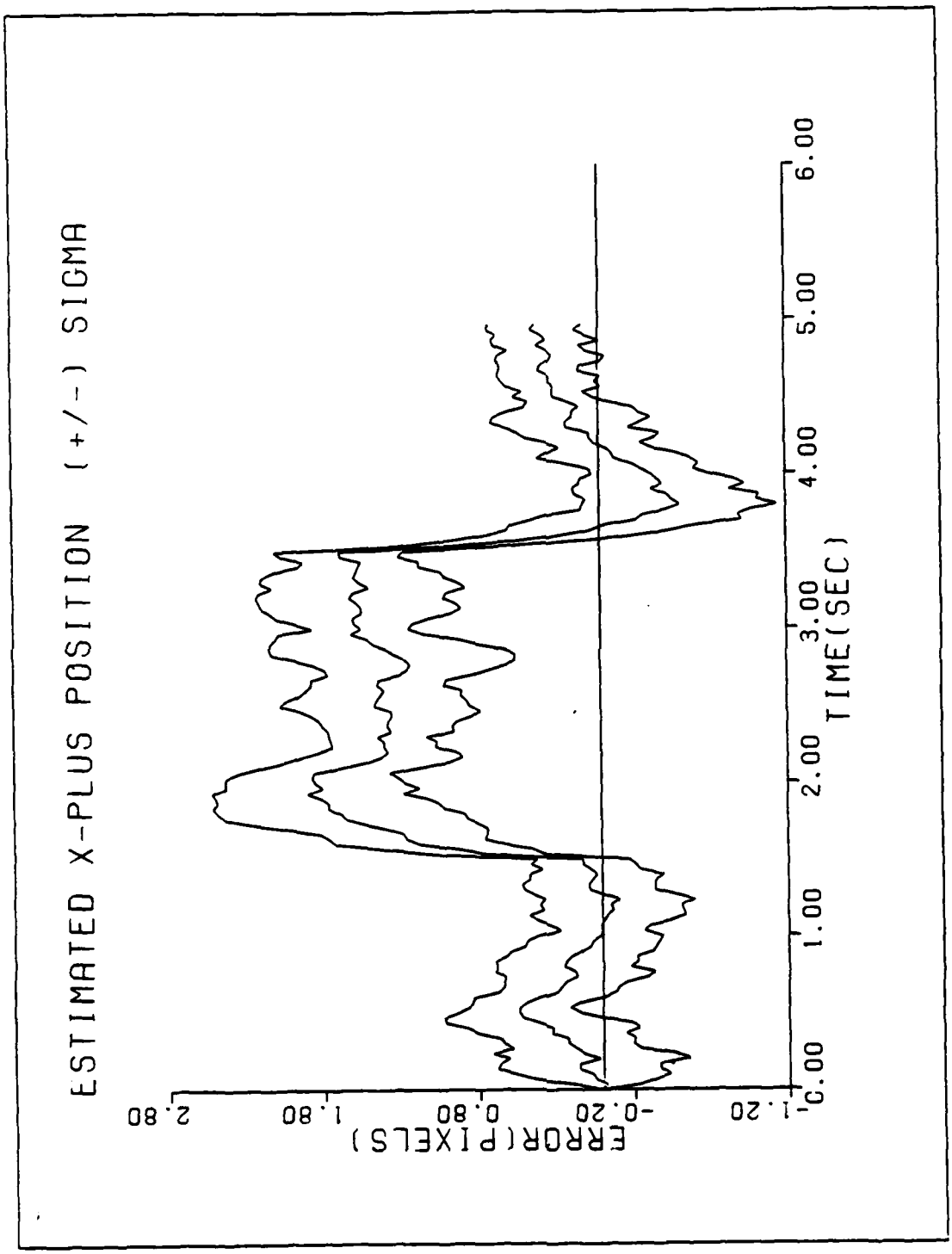


Figure A-8e. Performance Plot for T3GCM-BASE

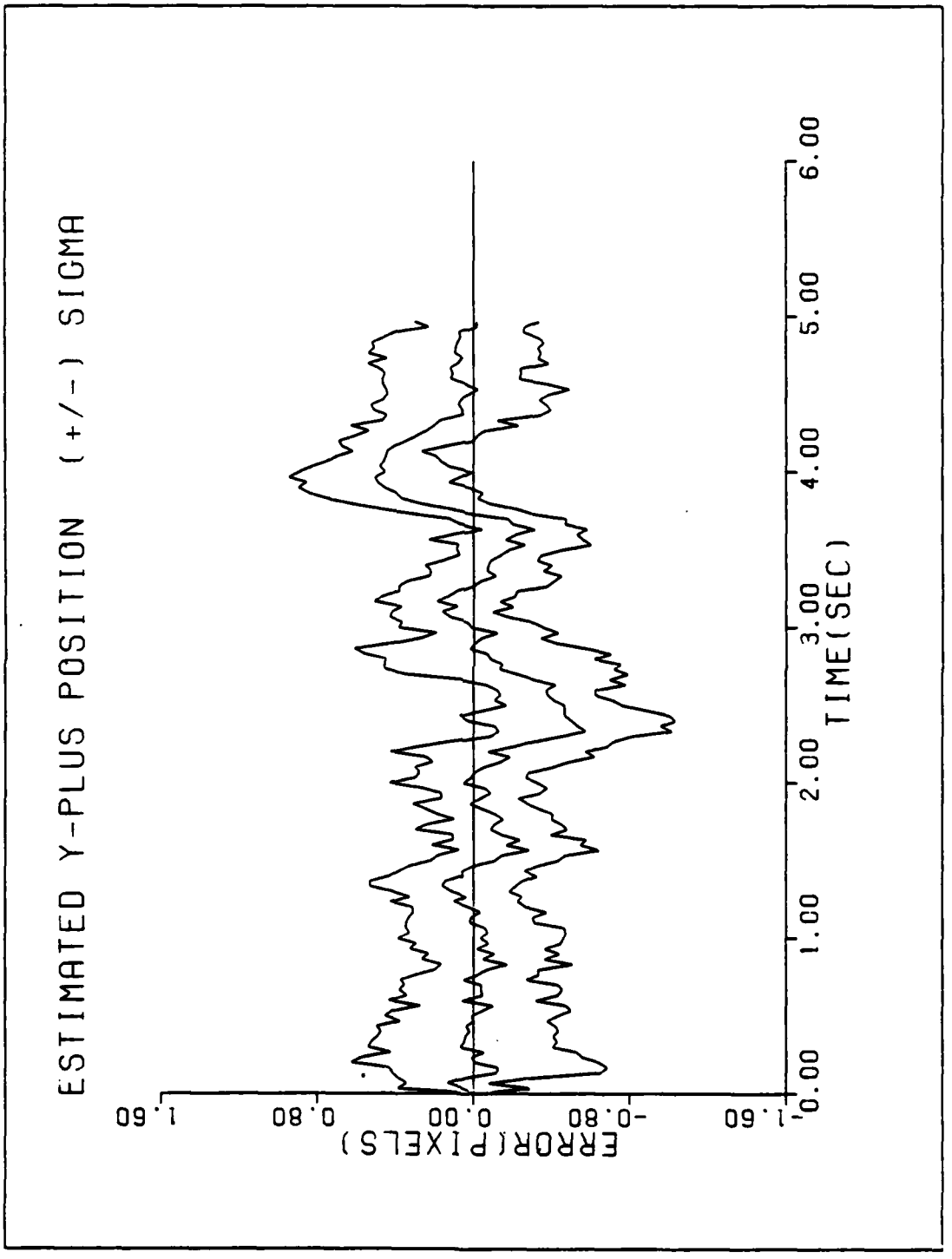


Figure A-8f. Performance Plot for T3GCM-BASE

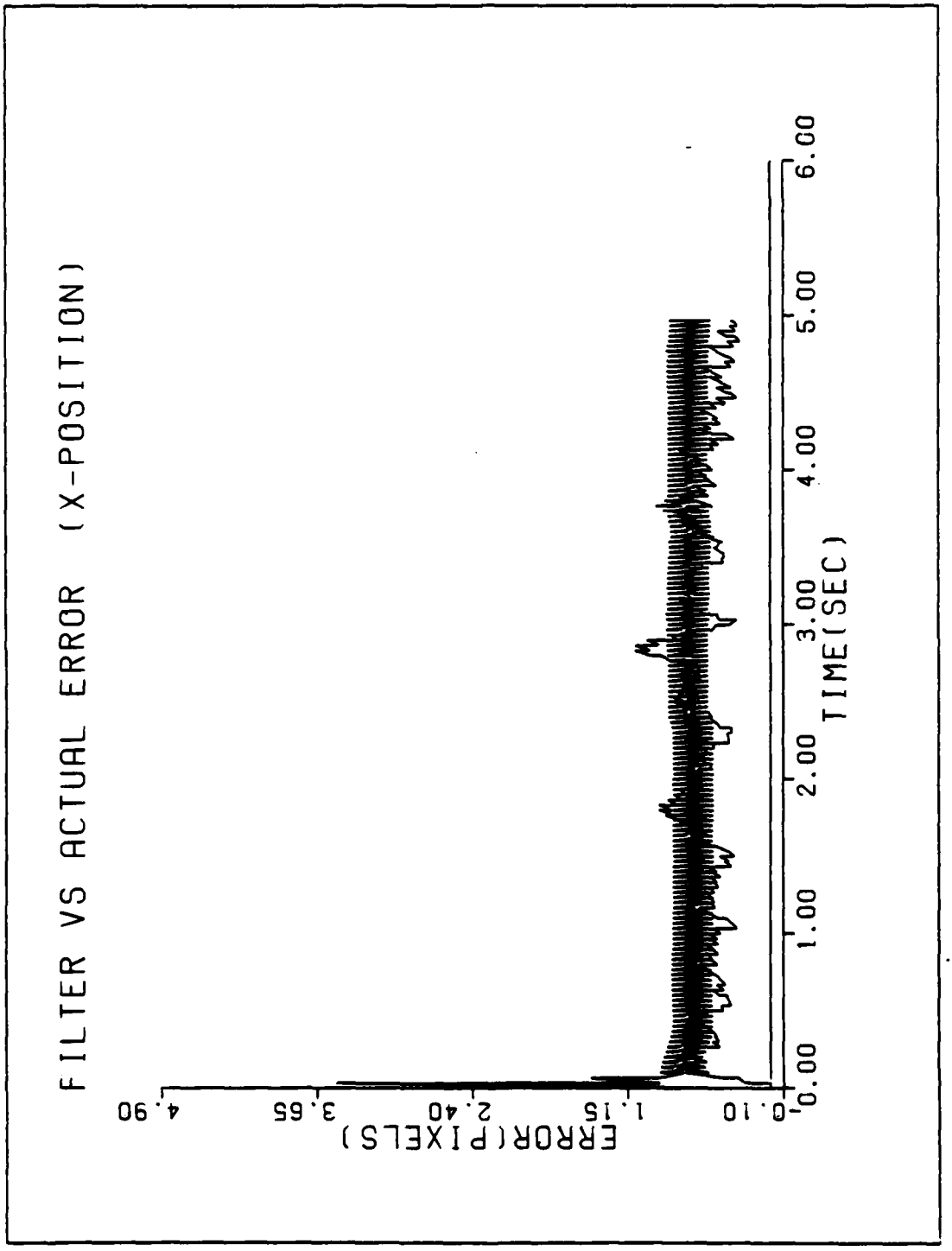


Figure A-9a. Performance Plot for T3G10MM-BASE

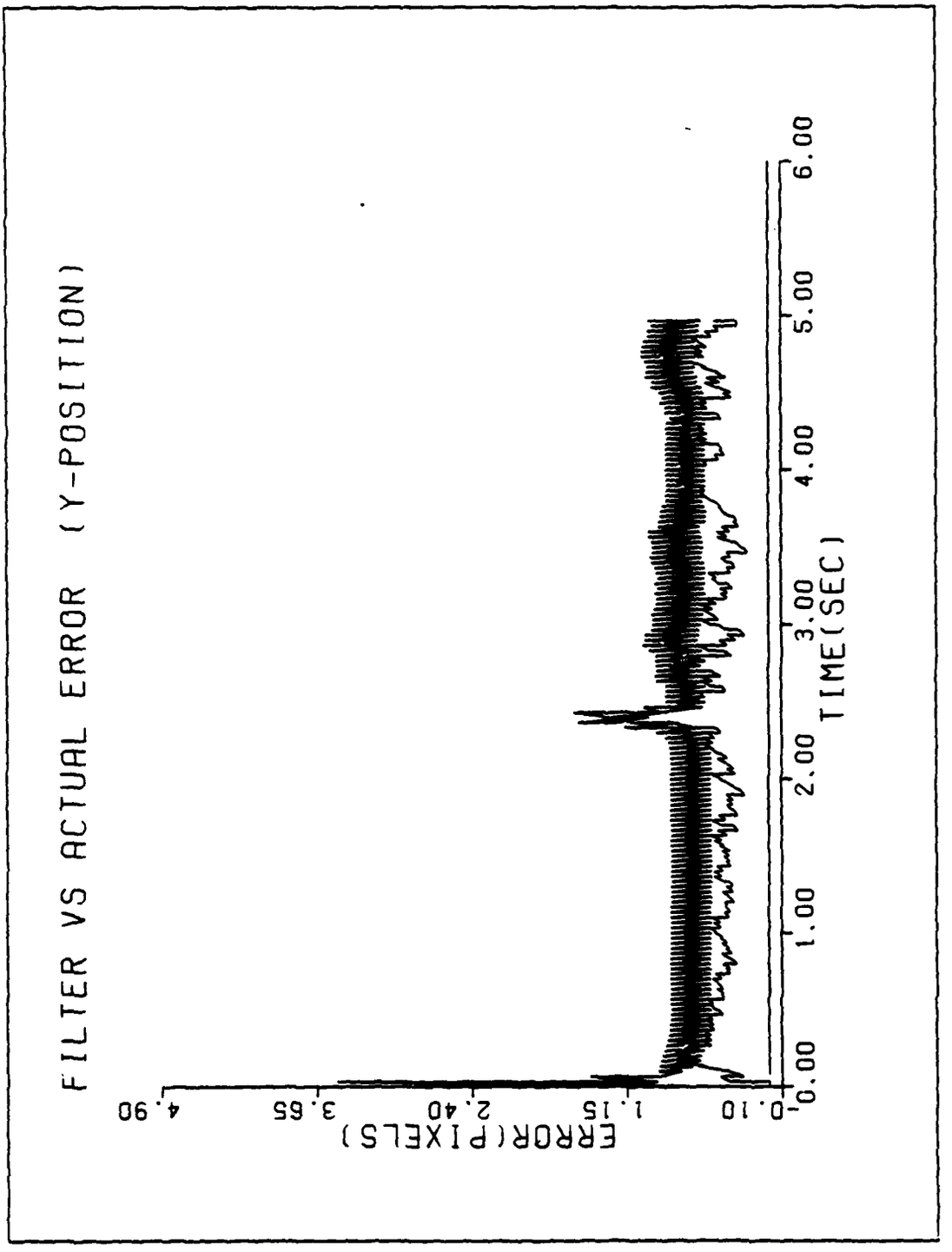


Figure A-9b. Performance Plot for T3G10MM-BASE

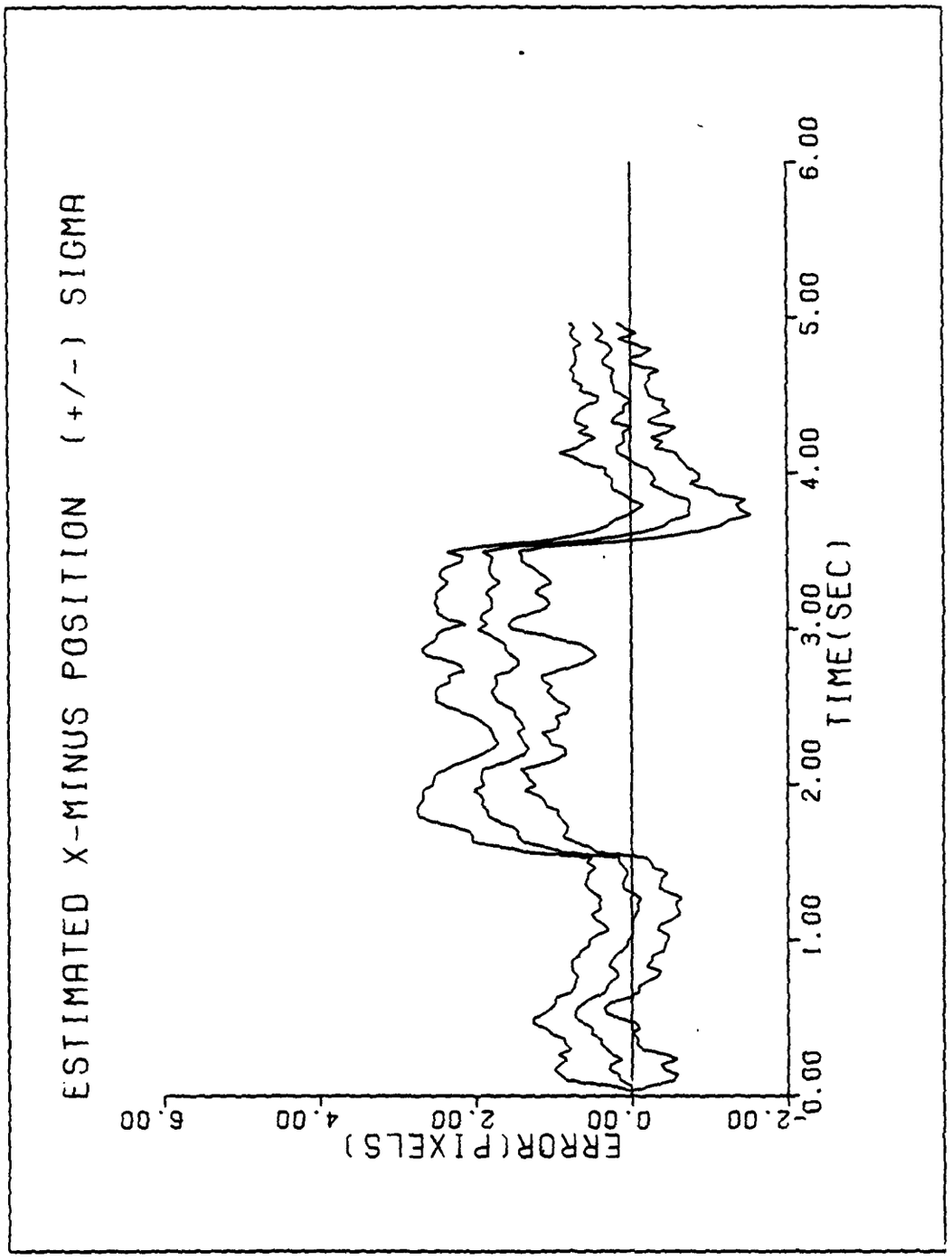


Figure A-9c. Performance Plot for T3G10NM-BASE

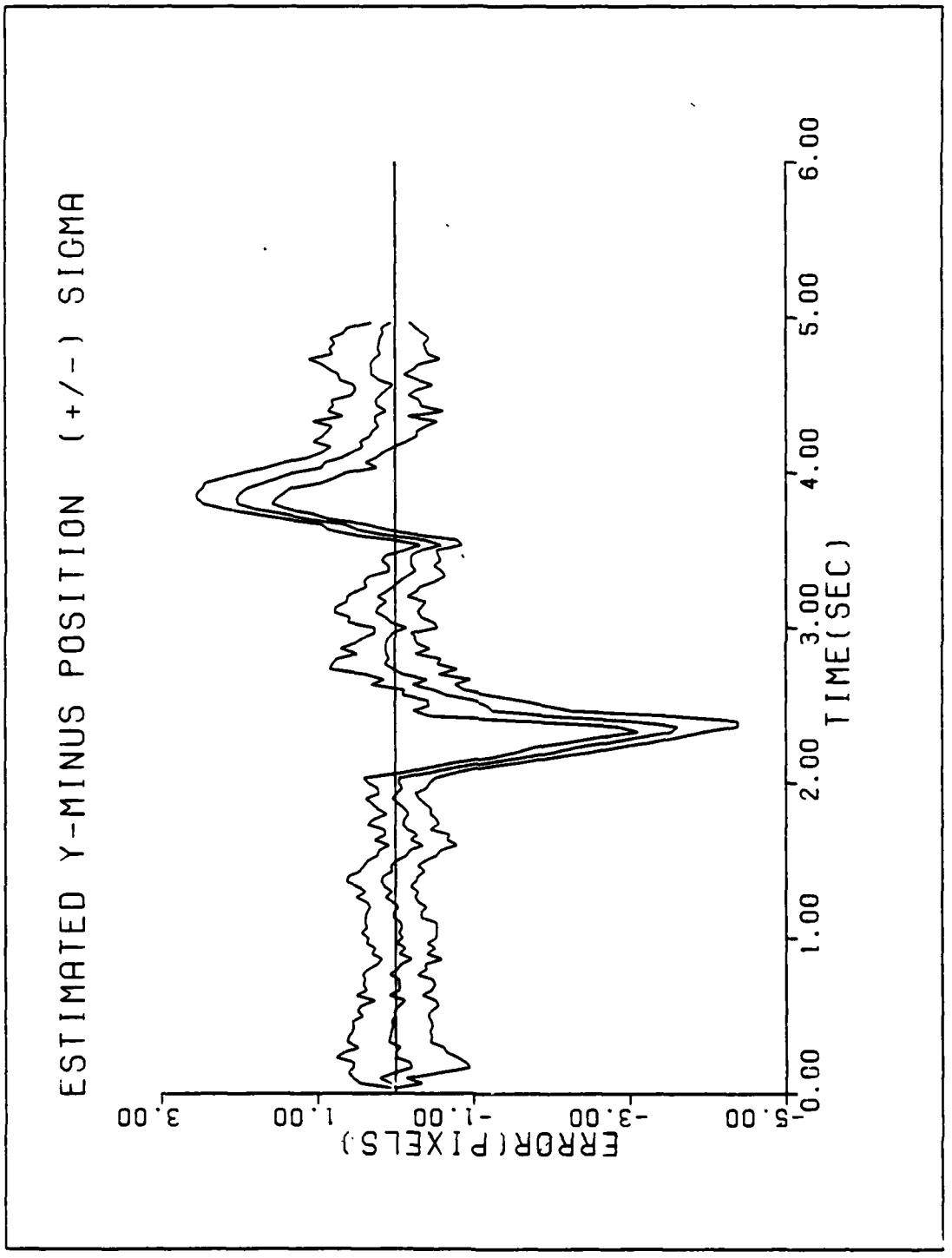


Figure A-9d. Performance Plot for TOGIOEM-BASE

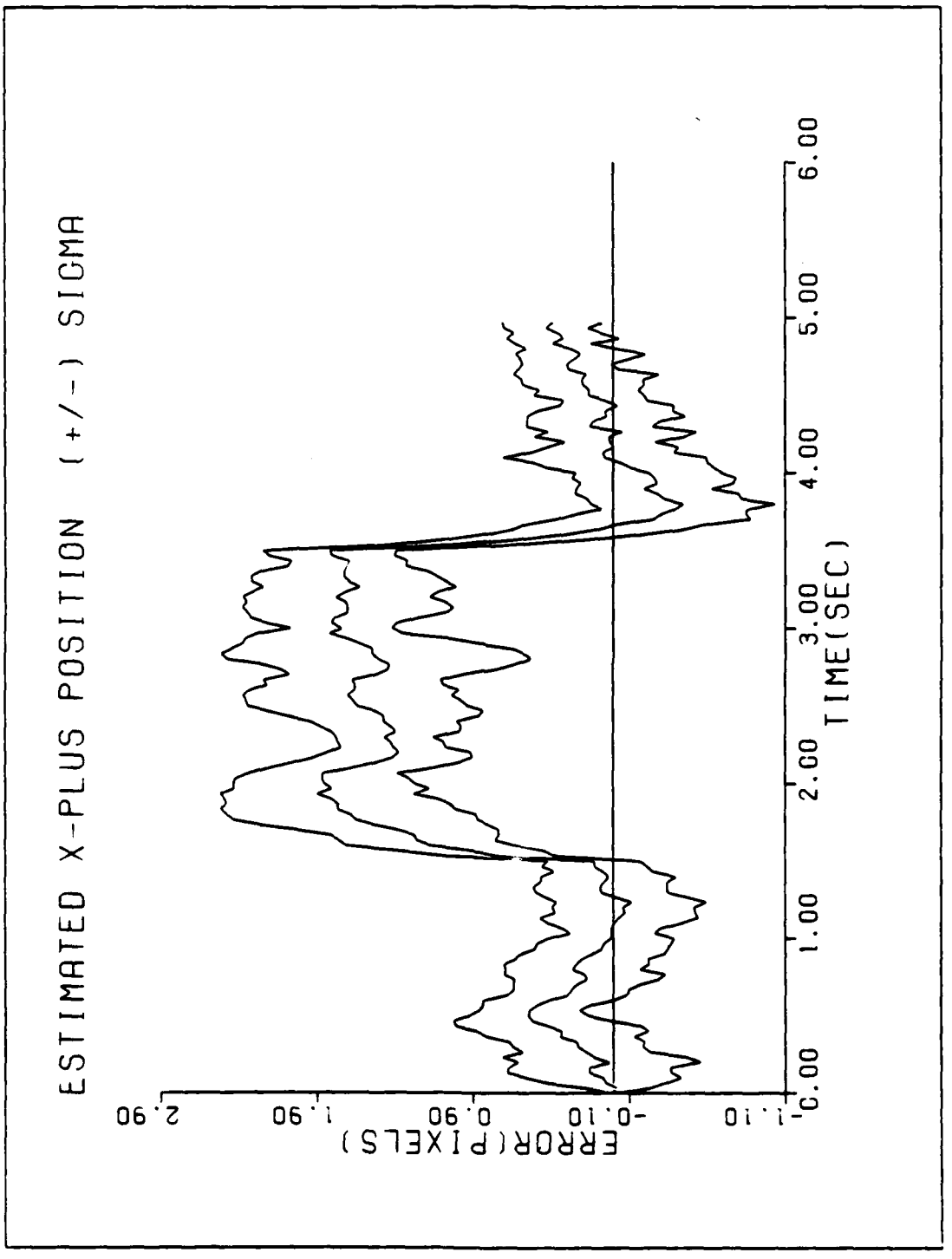


Figure A-9e. Performance Plot for T3G10:II-BASE

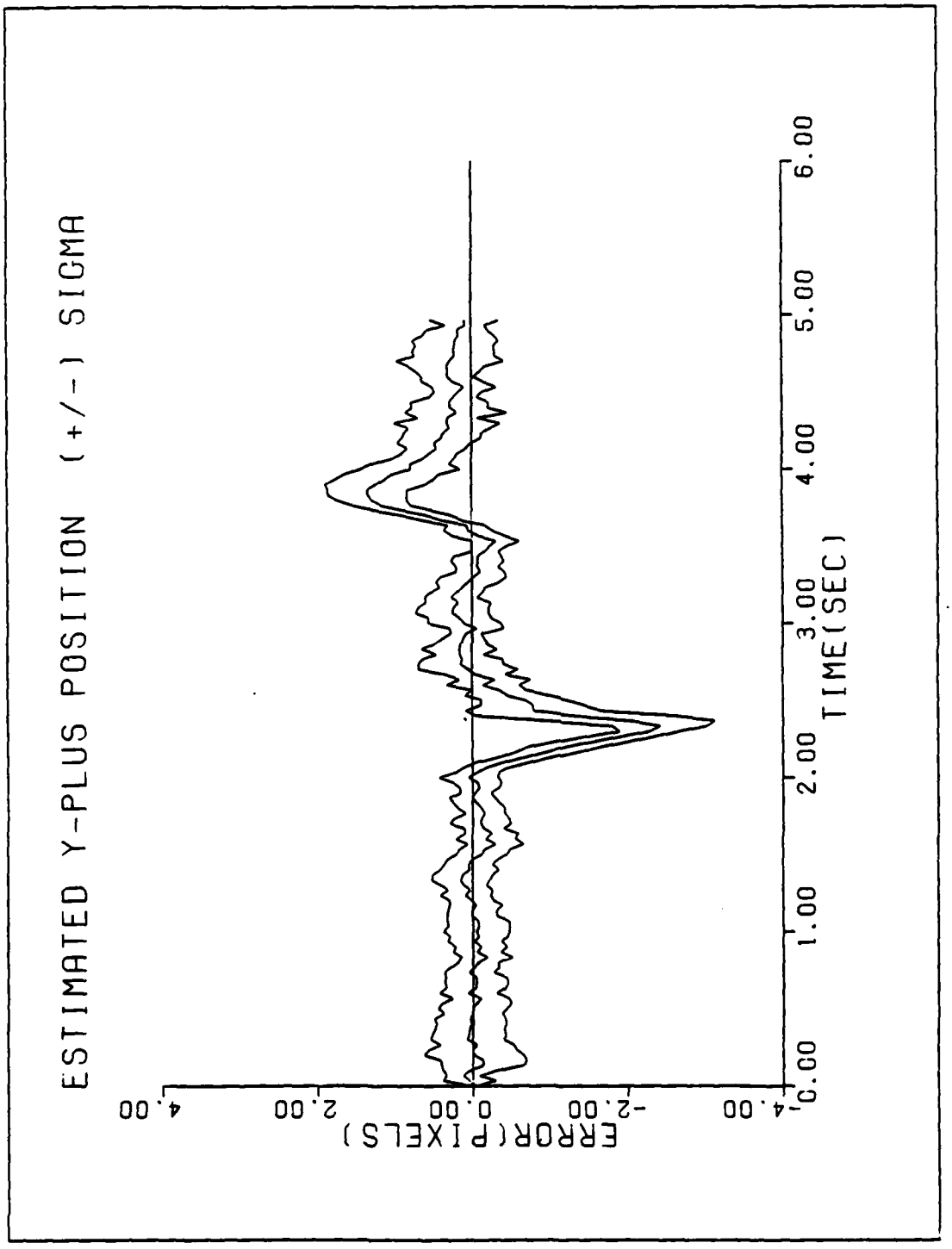


Figure A-9f. Performance Plot for T3G10MM-BASE

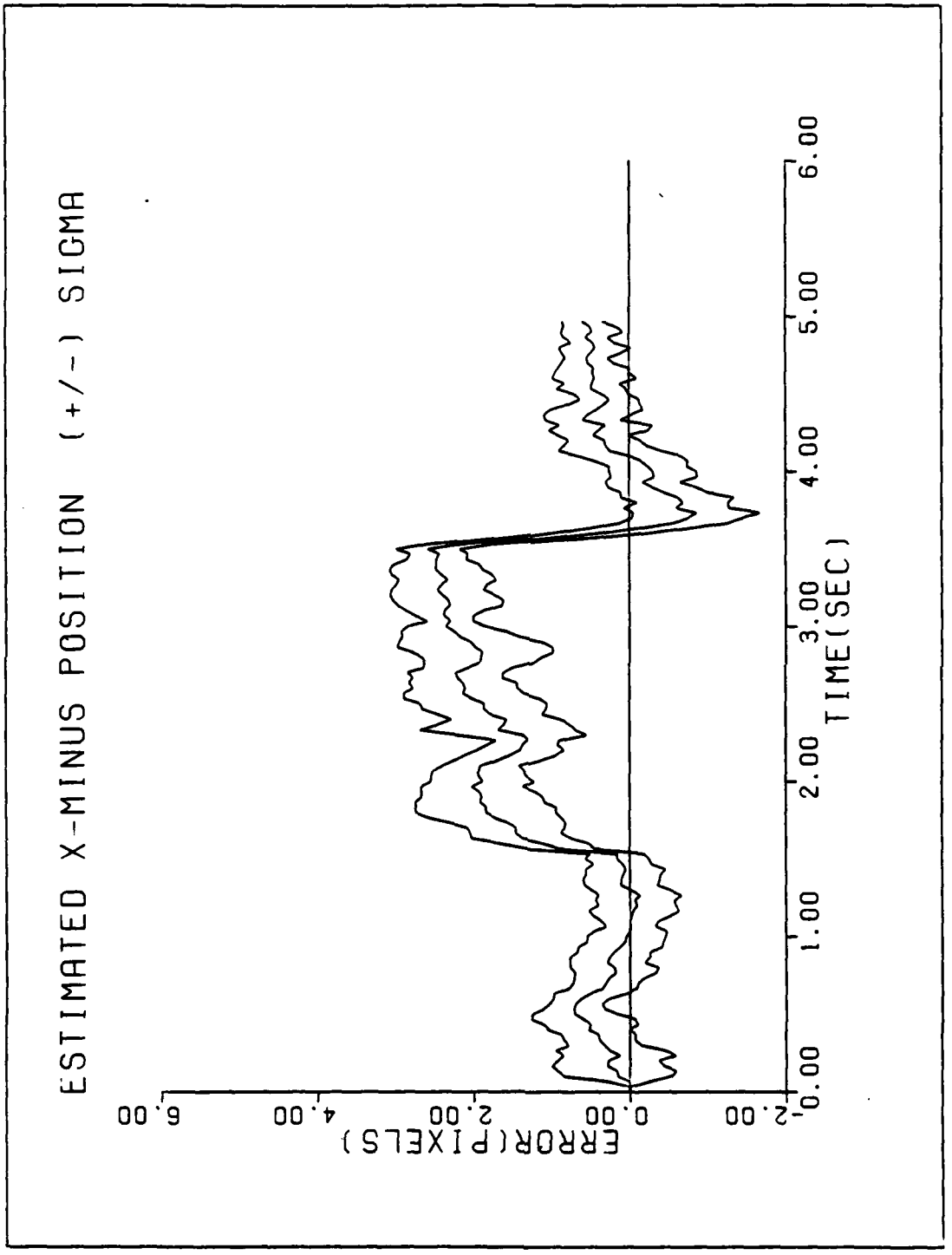


Figure A-10a. Performance Plot for T3620TT-BASE

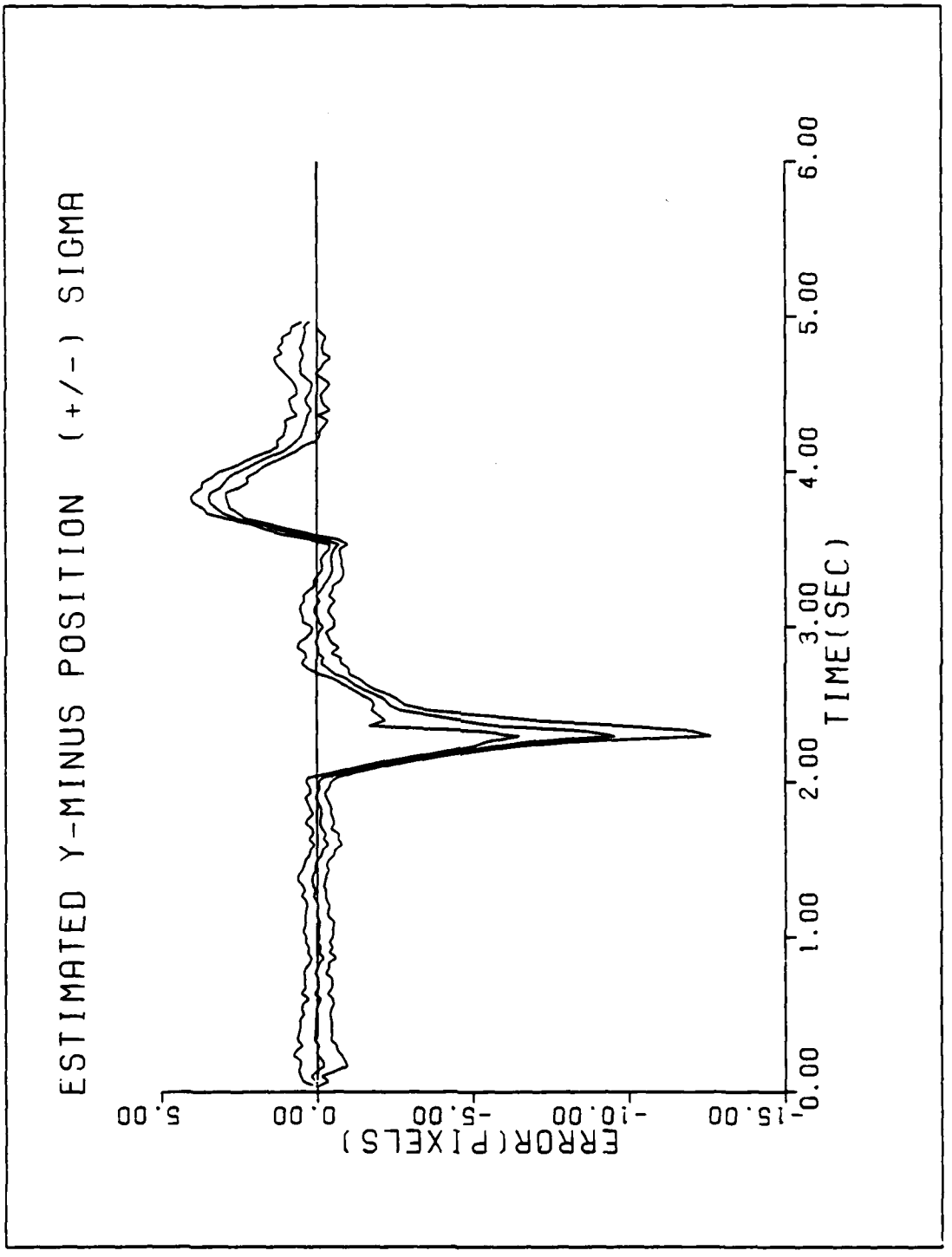


Figure A-10b. Performance Plot for TOGEOMI-BASE

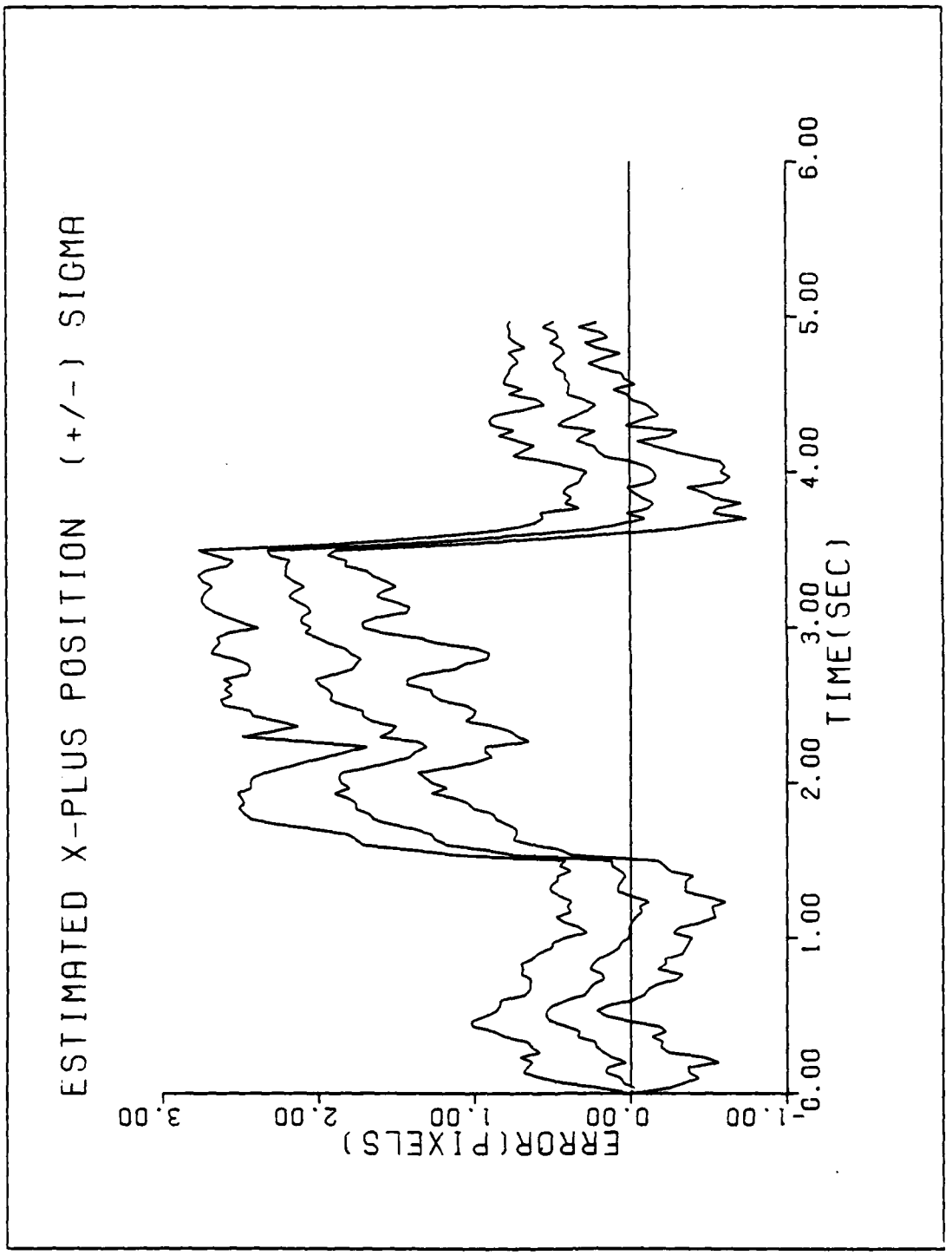


Figure A-10c. Performance Plot for T3GE0'II-BASE

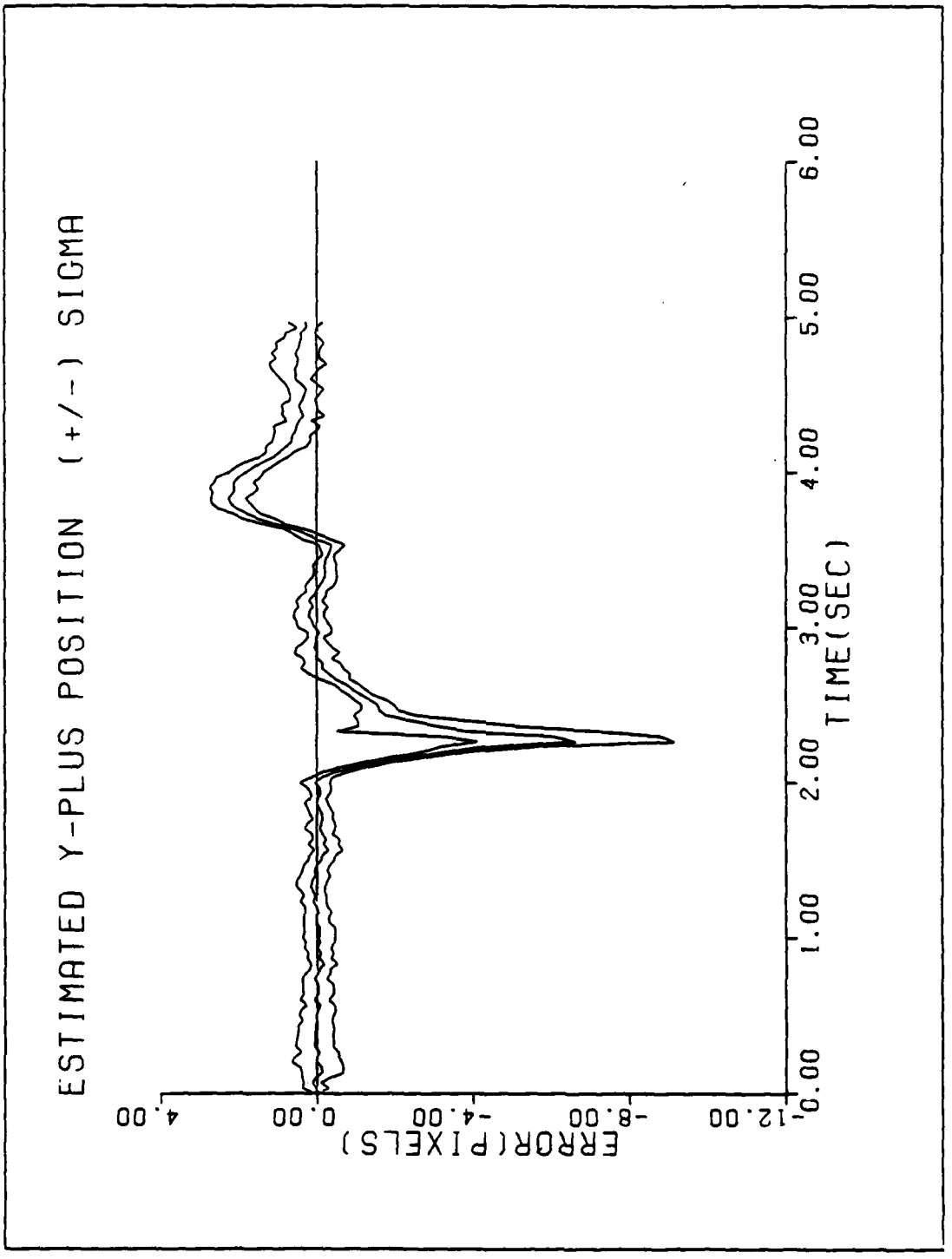


Figure A-10d. Performance Plot for TOGGOTT-BASE

APPENDIX B

Performance Plots for the Bayesian MMAF

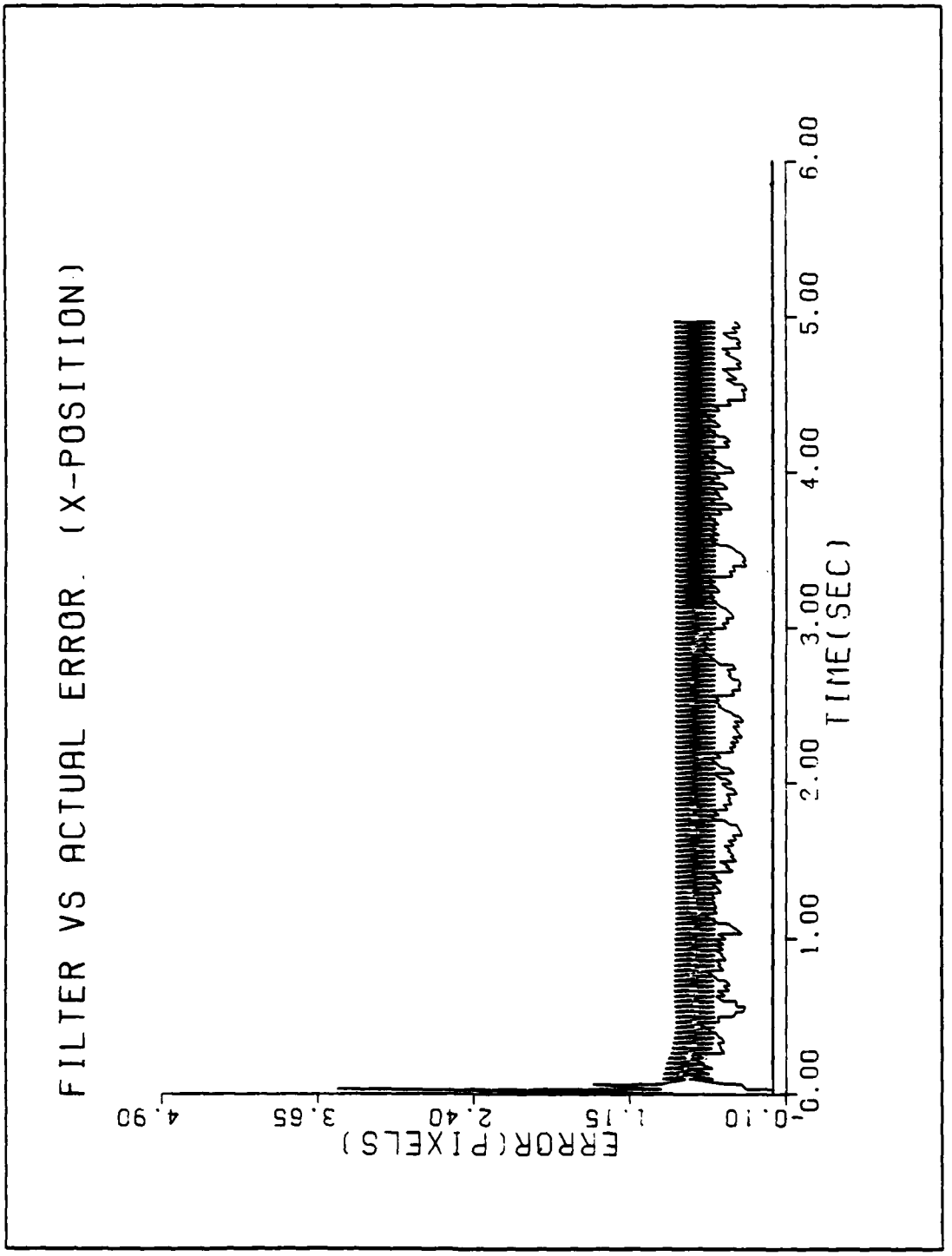


Figure B-1a. Performance Plot for TEM-BAY

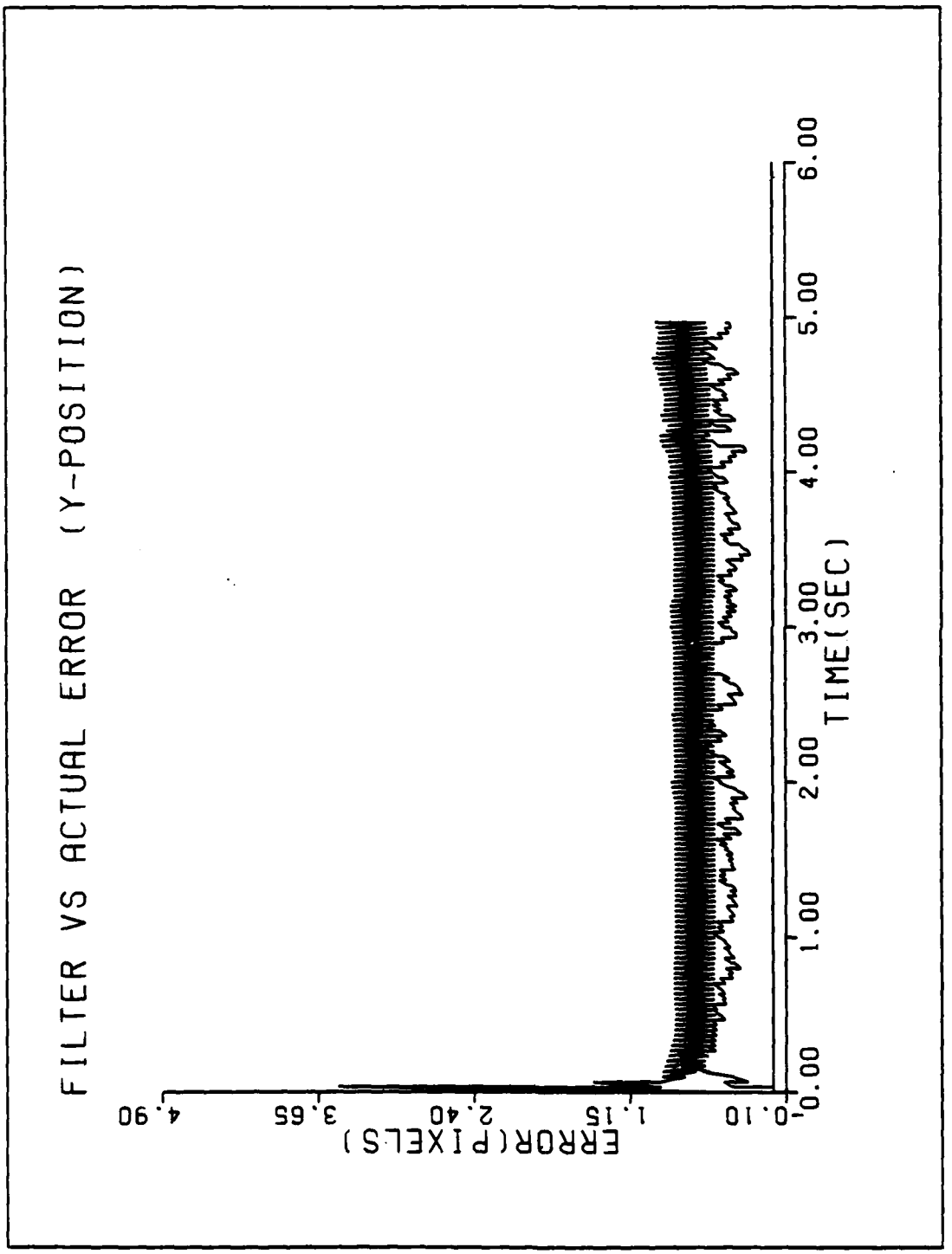


Figure B-1b. Performance Plot for TIME-BAY

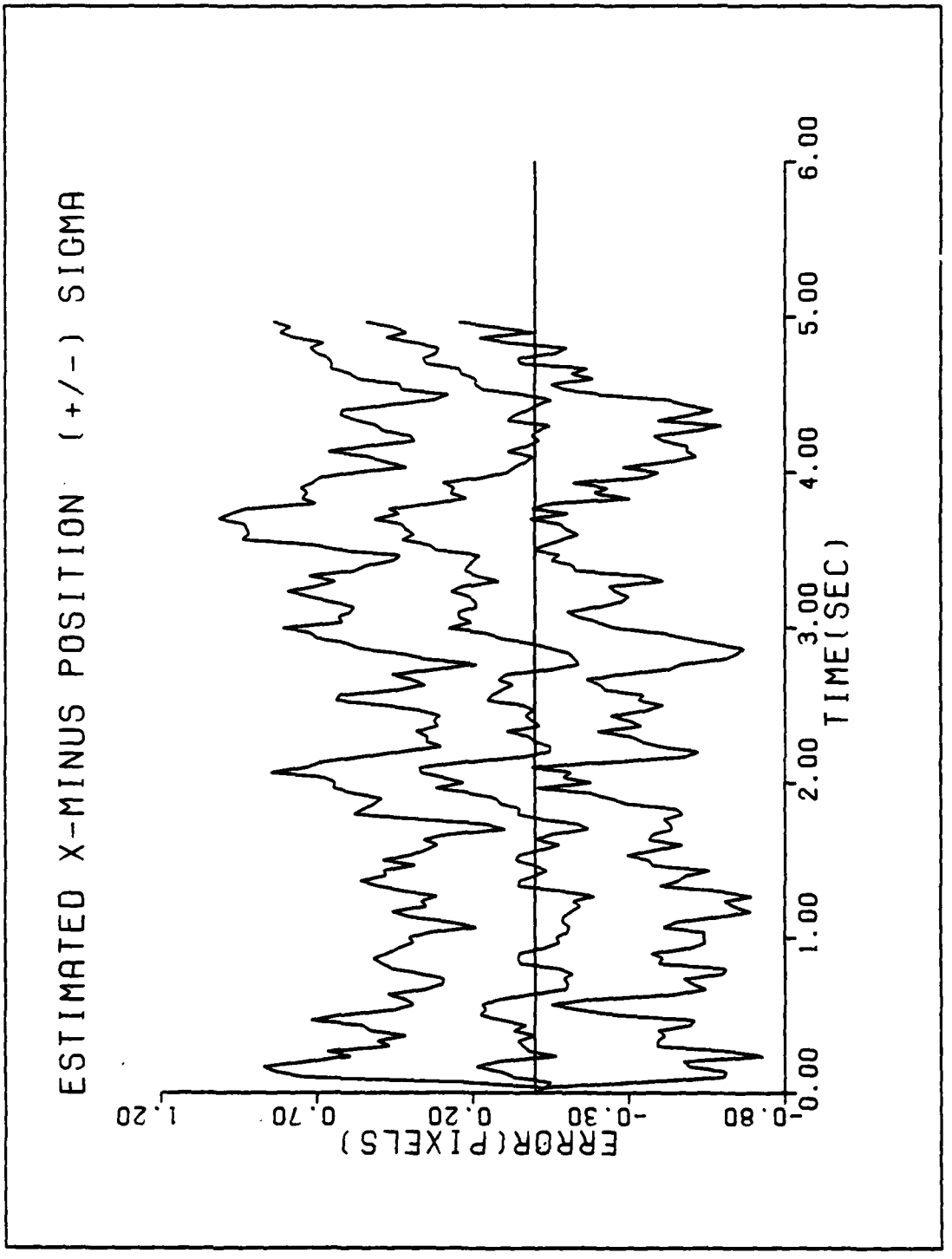


Figure B-1c. Performance Plot for TIM-BAY

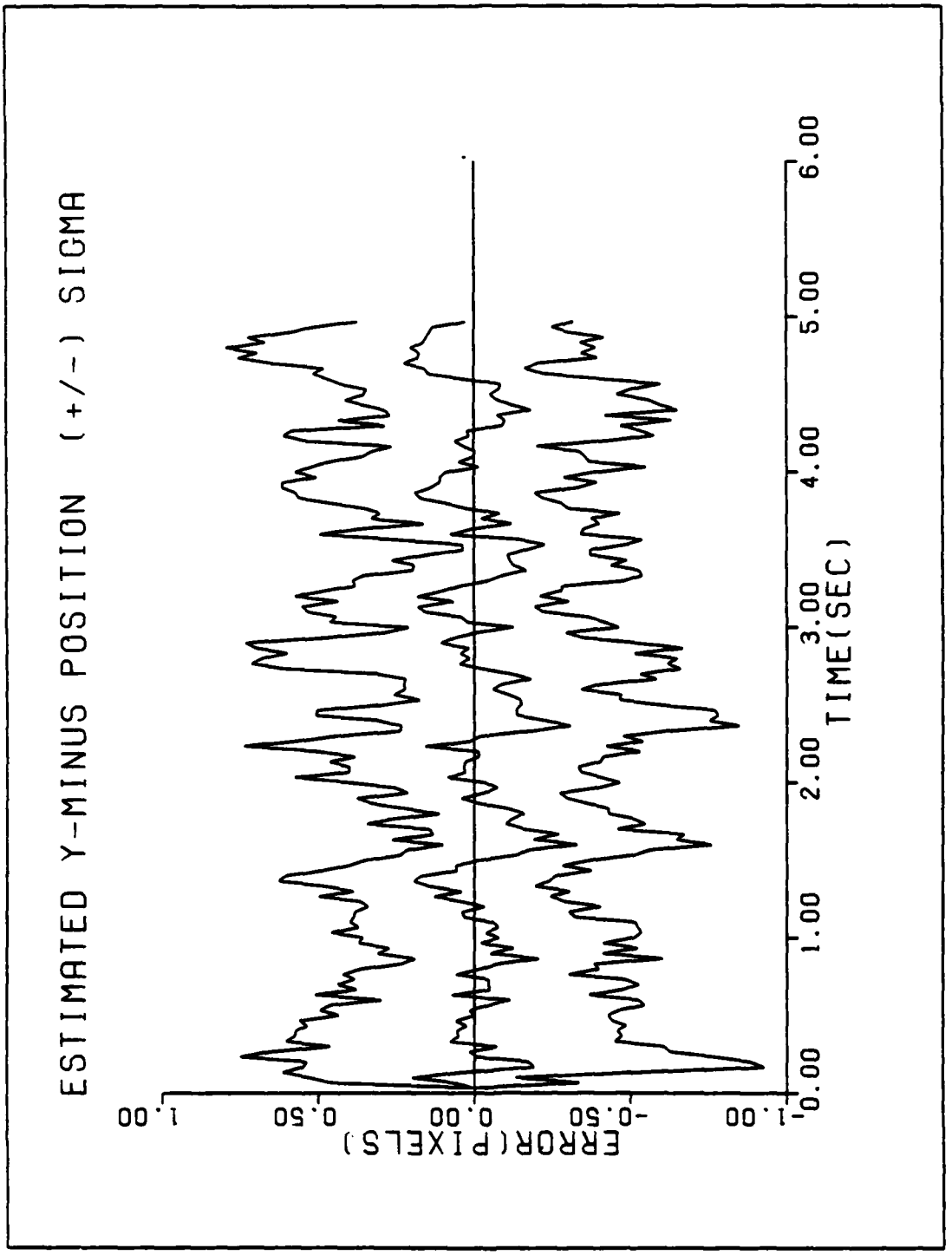


Figure B-1d. Performance Plot for TIME-BAY

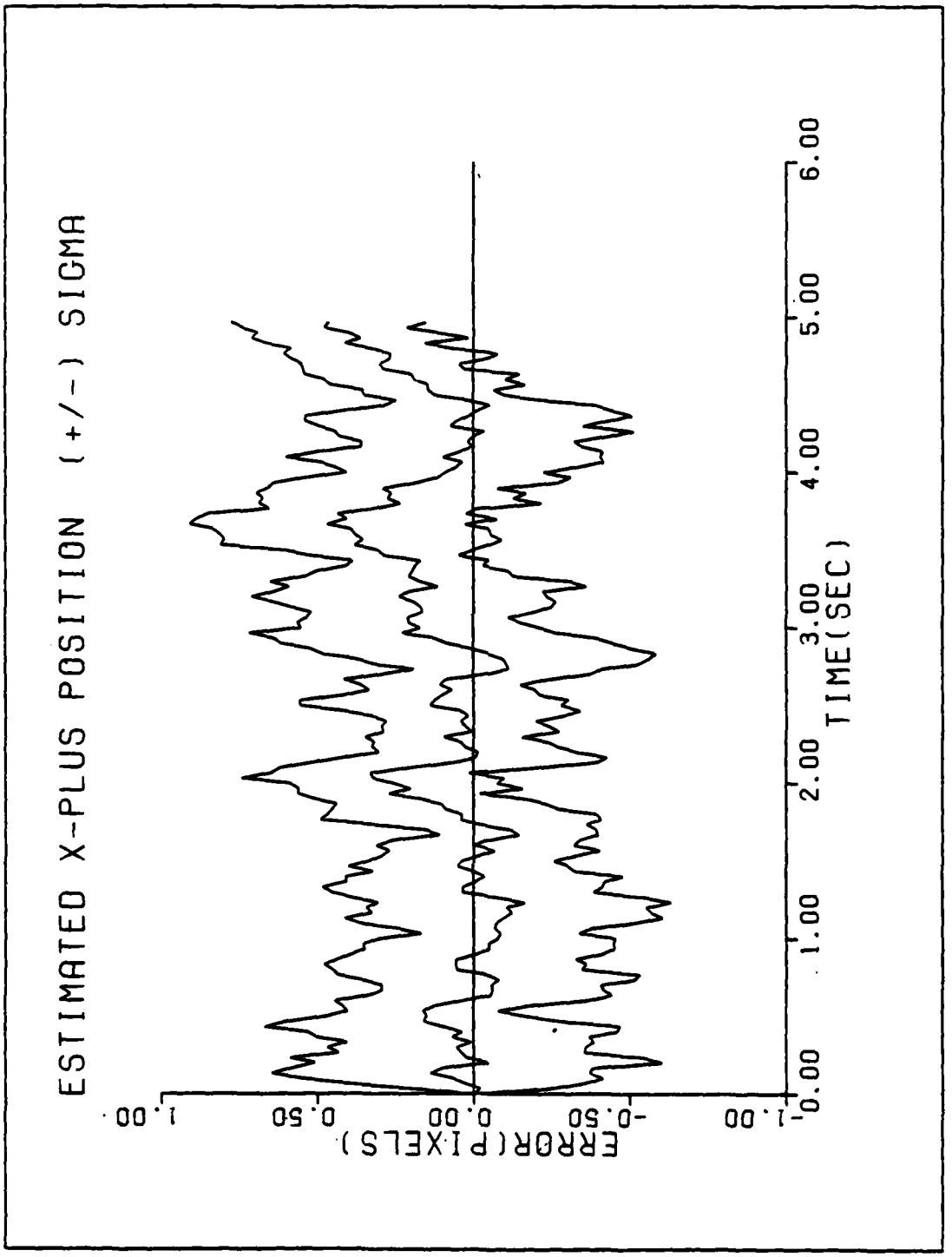


Figure B-1e. Performance Plot for TIME-BAY

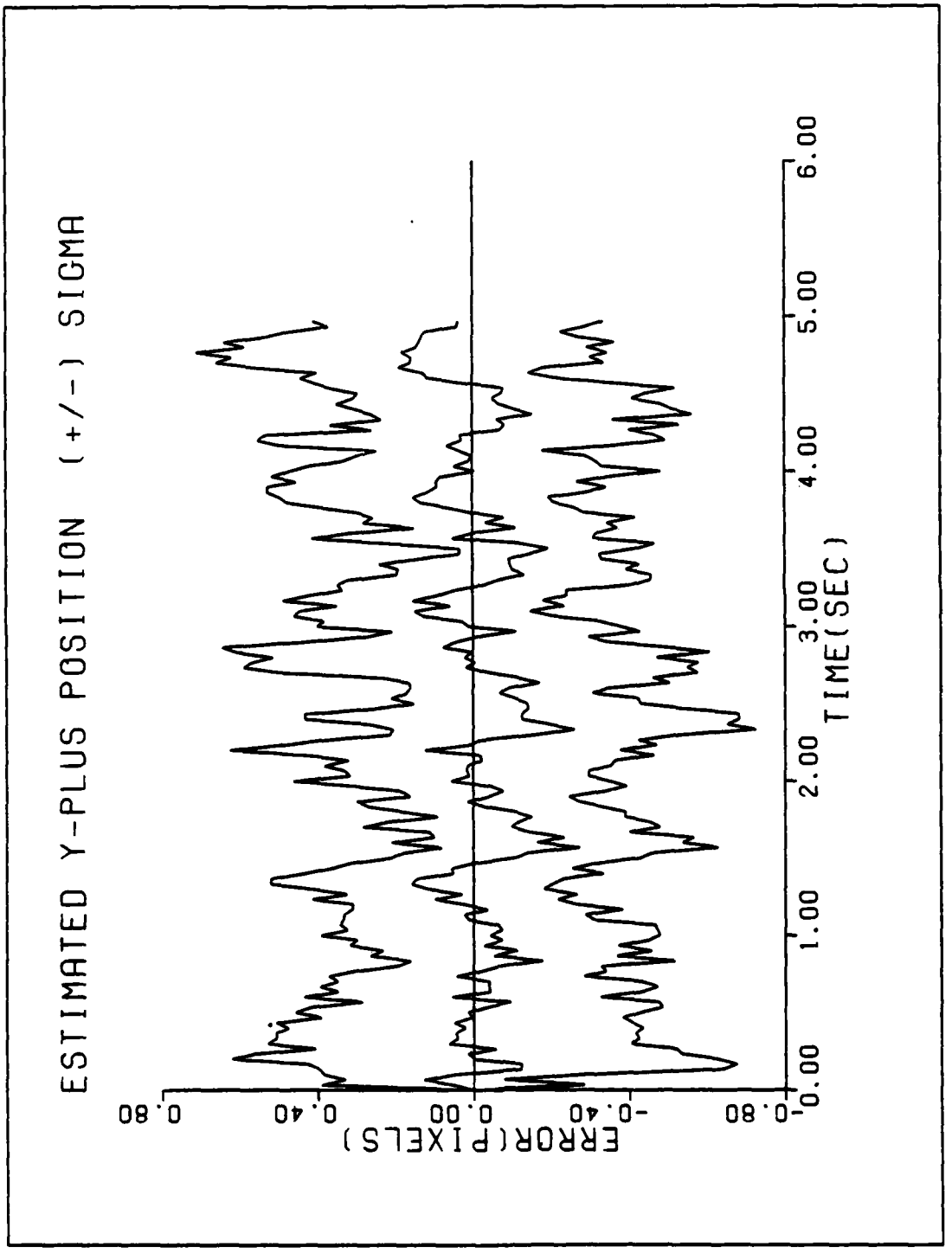


Figure B-1f. Performance Plot for TIMI-BAY

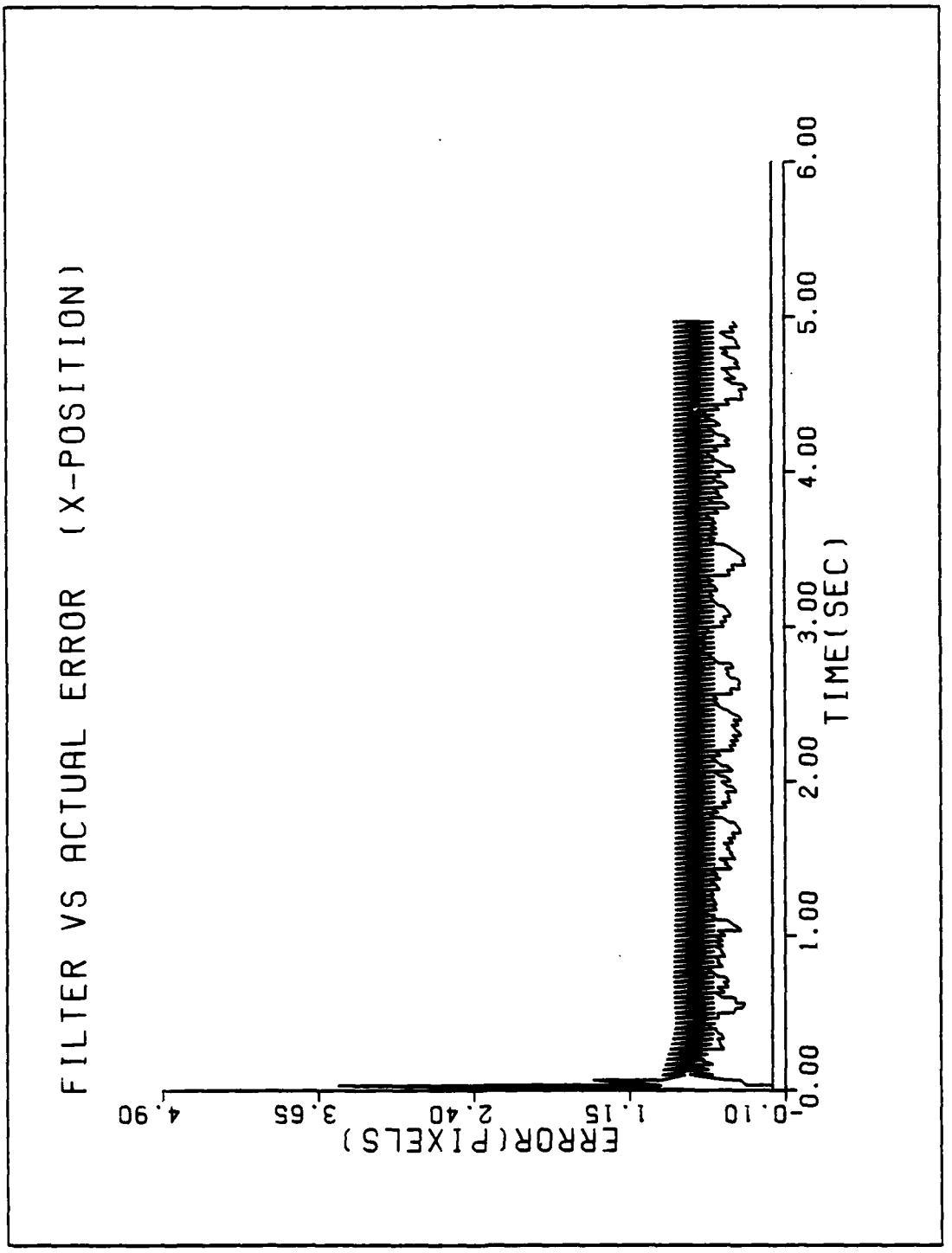


Figure B-2a. Performance Plot for TIFI-BAY

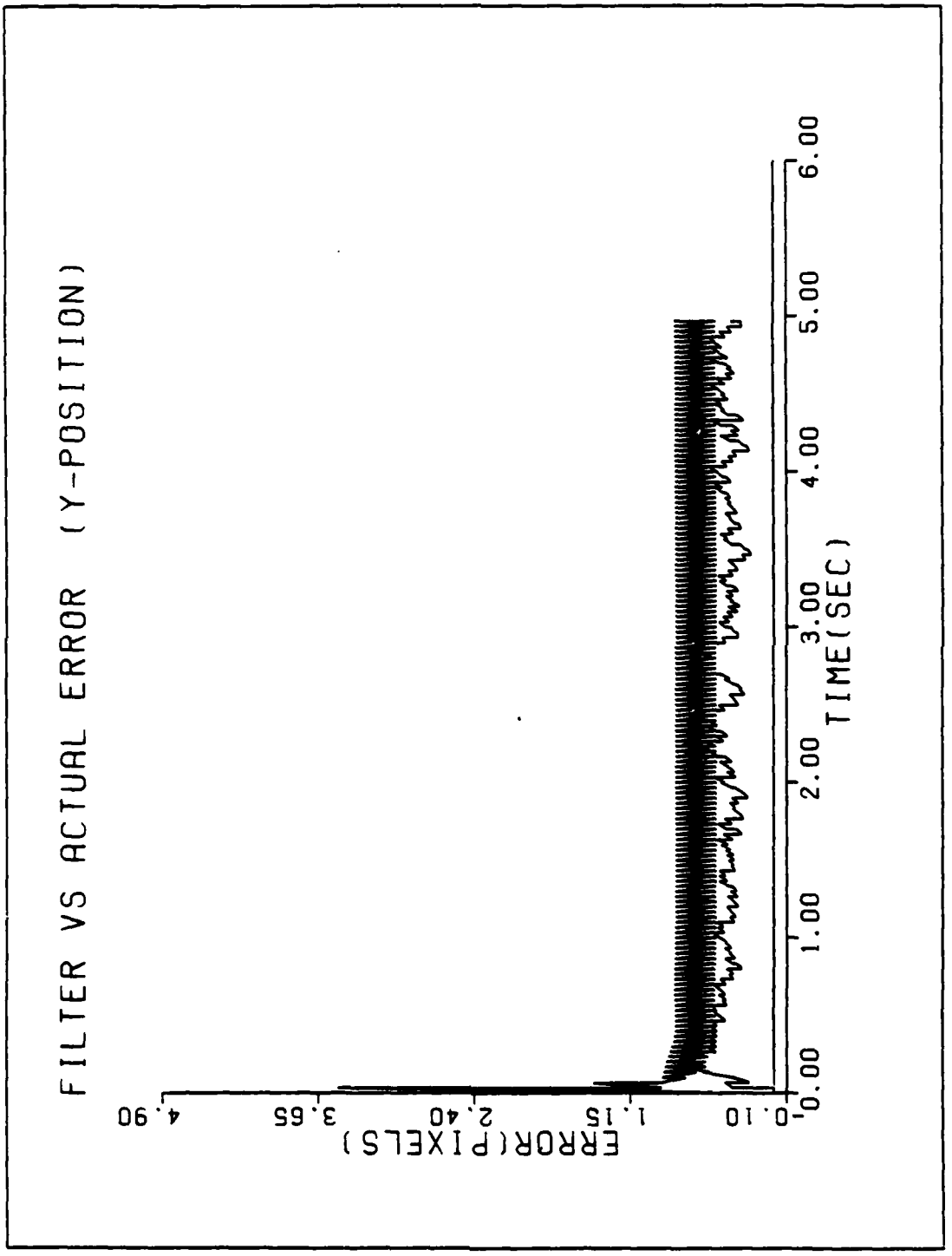


Figure B-2b. Performance Plot for TIFI-BAY

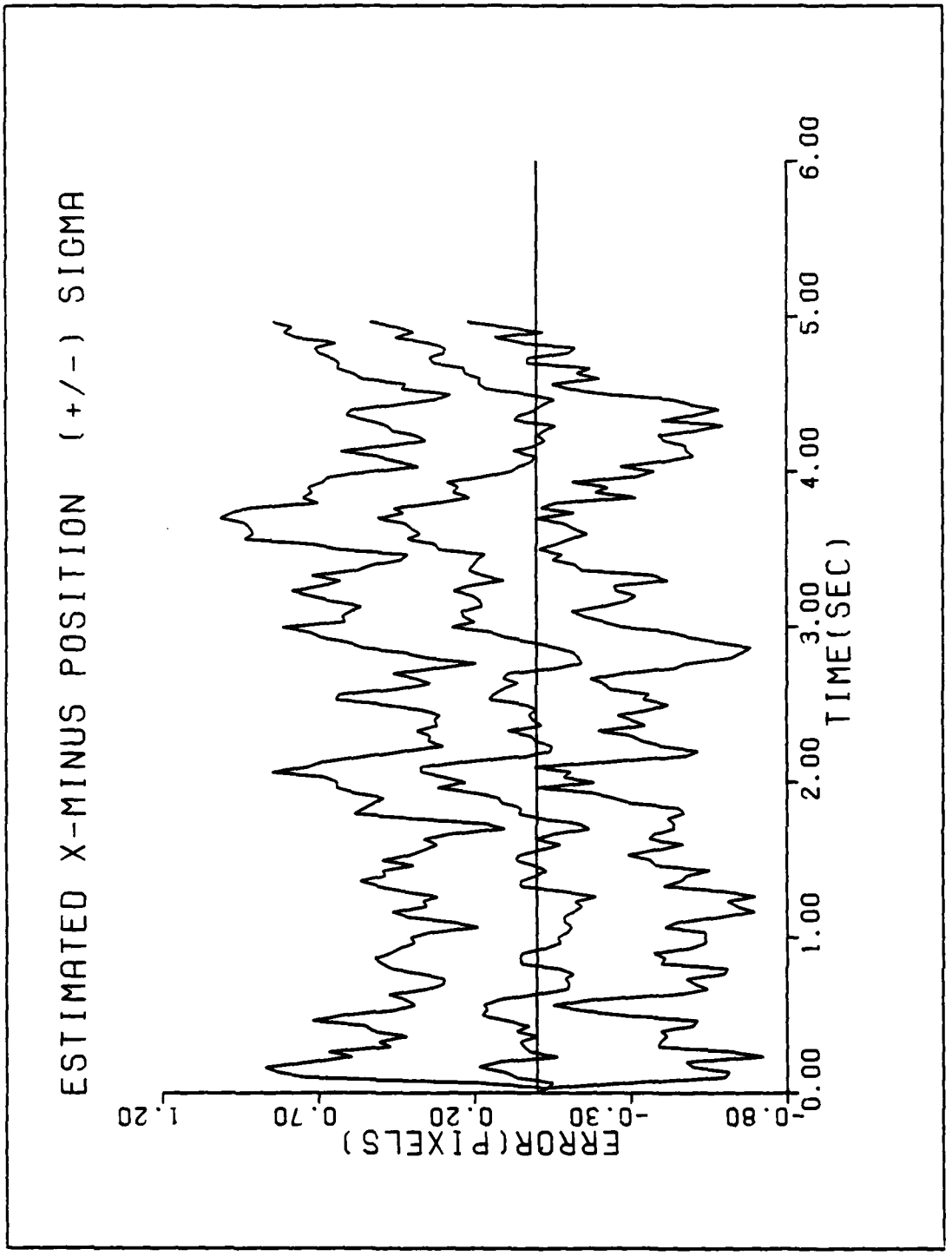


Figure B-2c. Performance Plot for TIFI-BAY

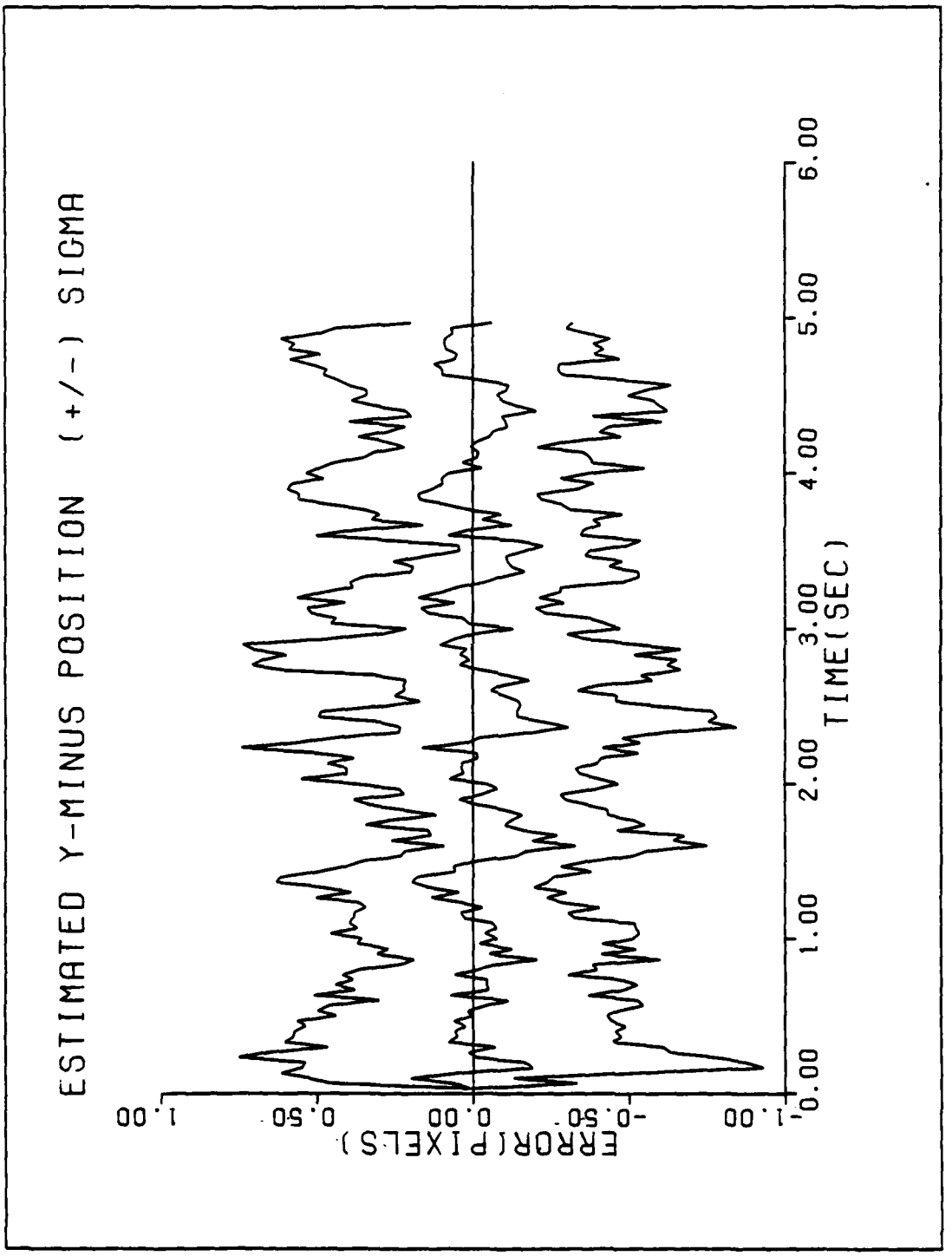


Figure B-2d. Performance Plot for TIFI-BAY

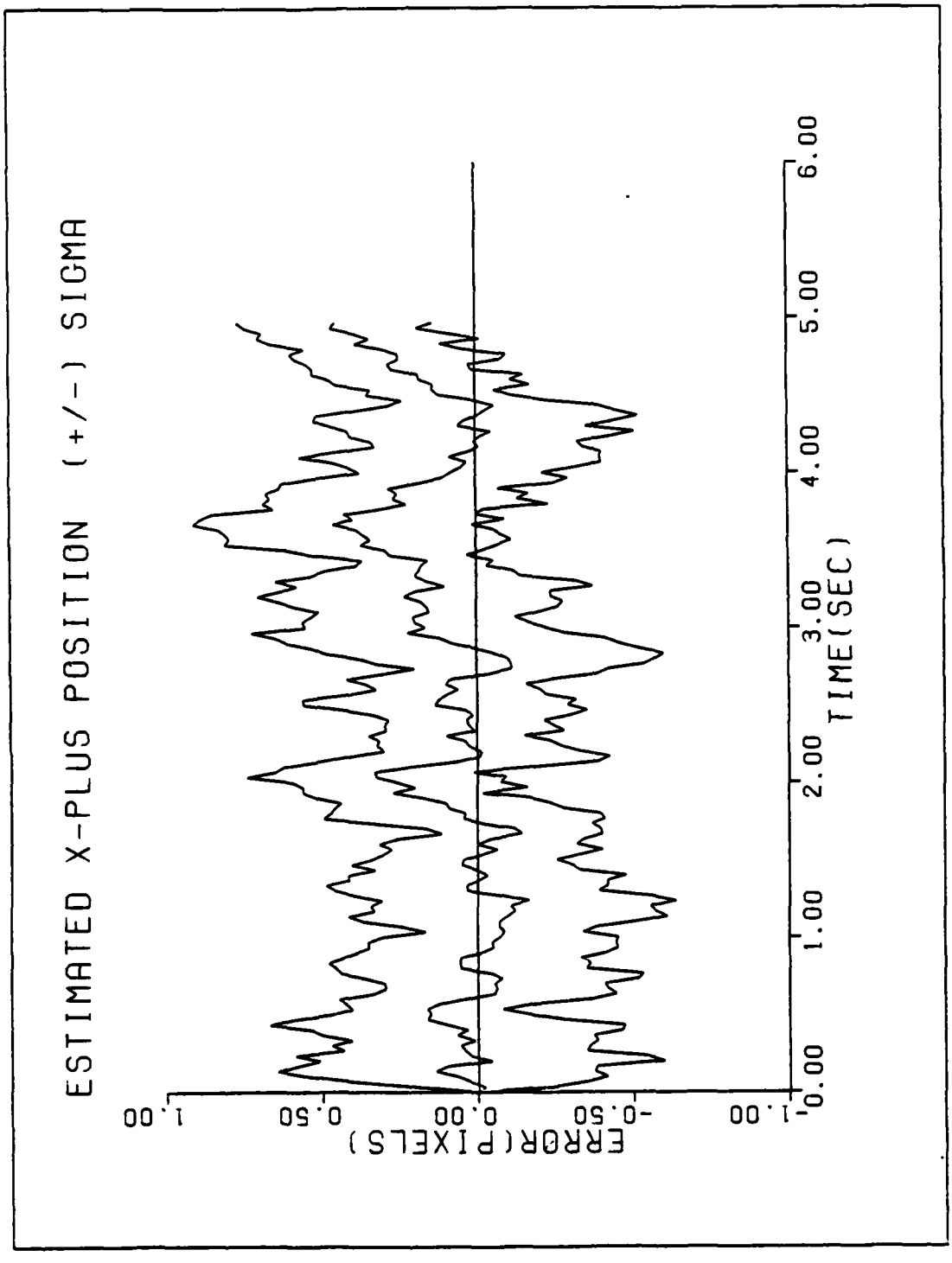


Figure B-2e. Performance Plot for TIFI-BAY

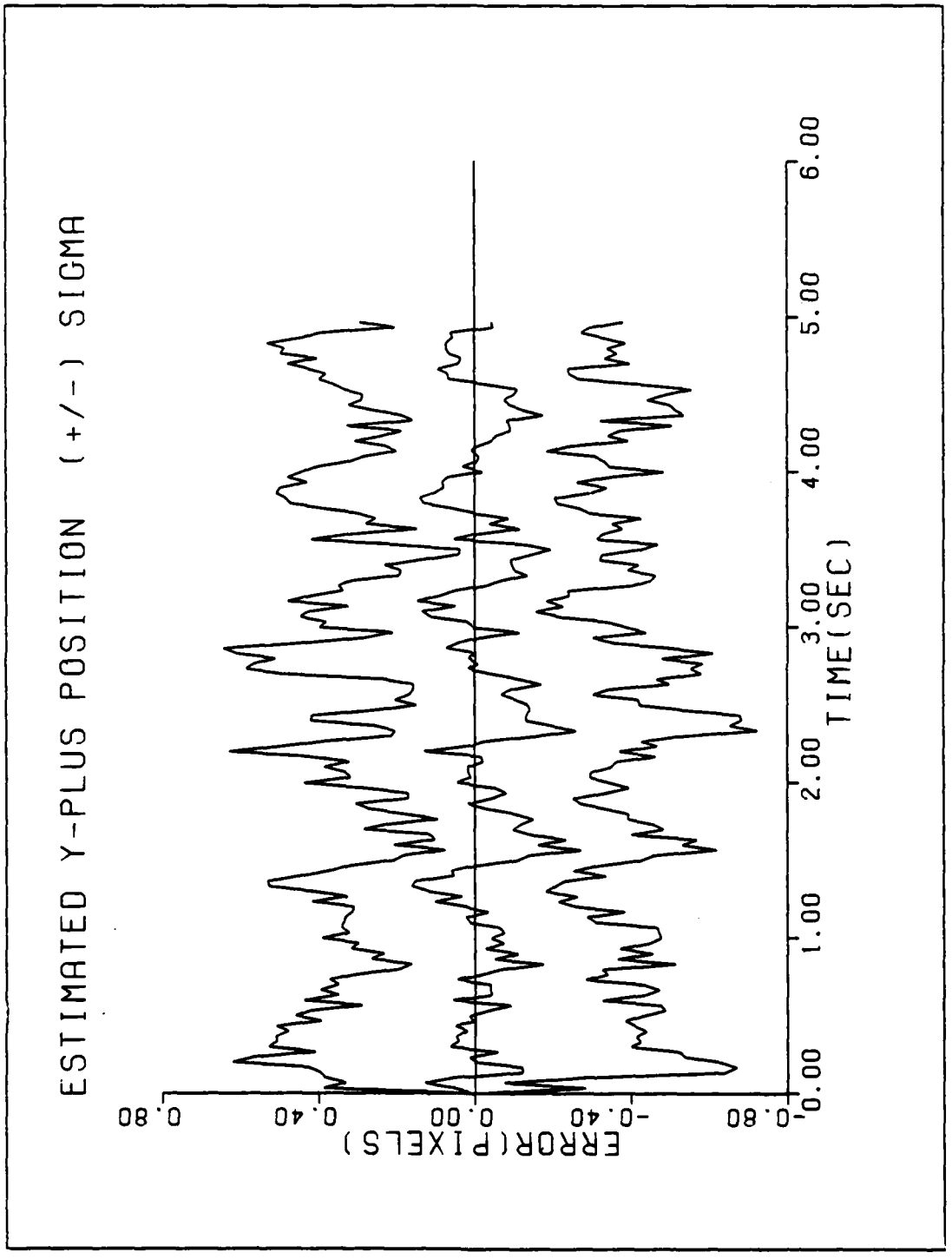


Figure B-2f. Performance Plot for TIFI-BAY

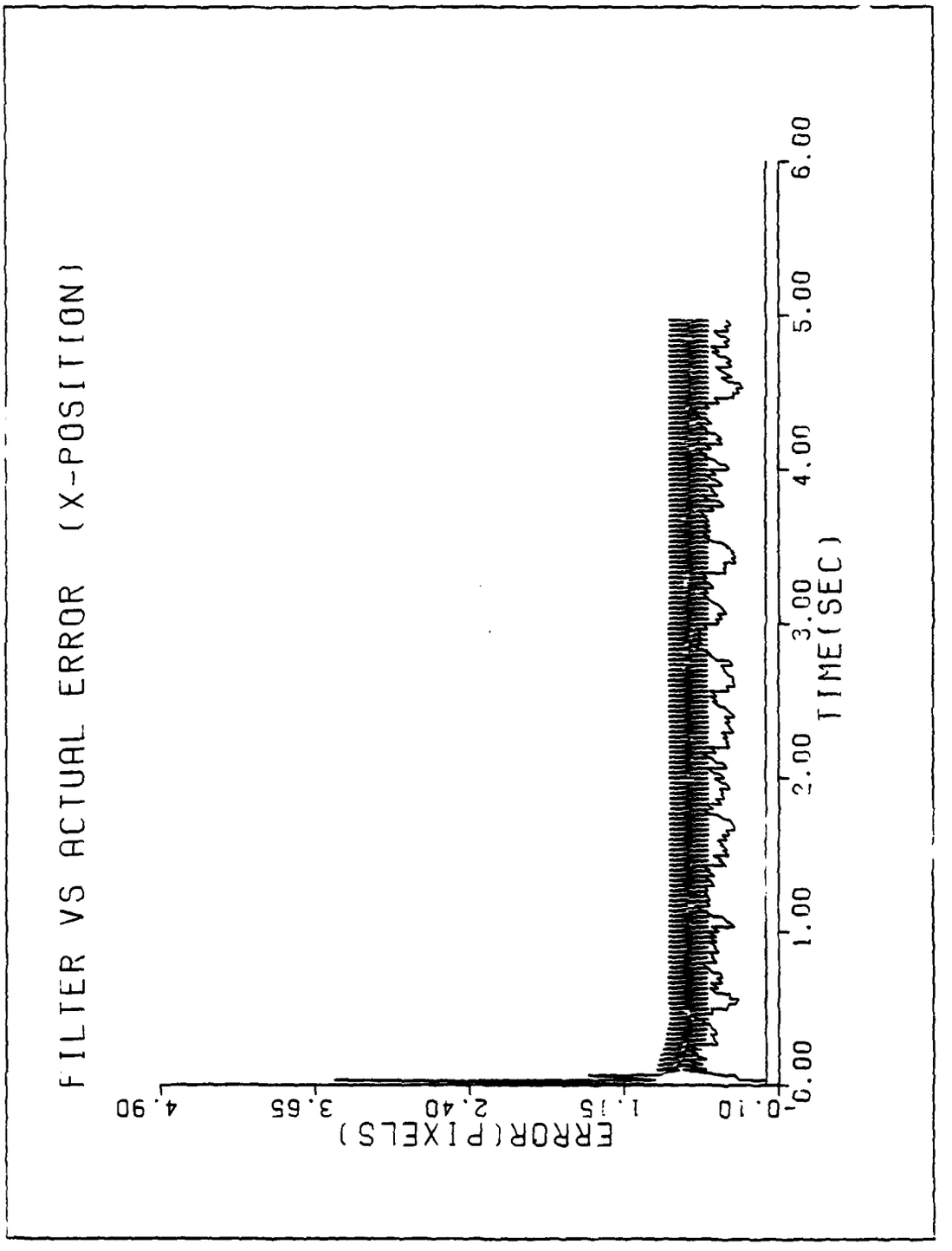


Figure B-3a. Performance Plot for T2GMMI-BAY

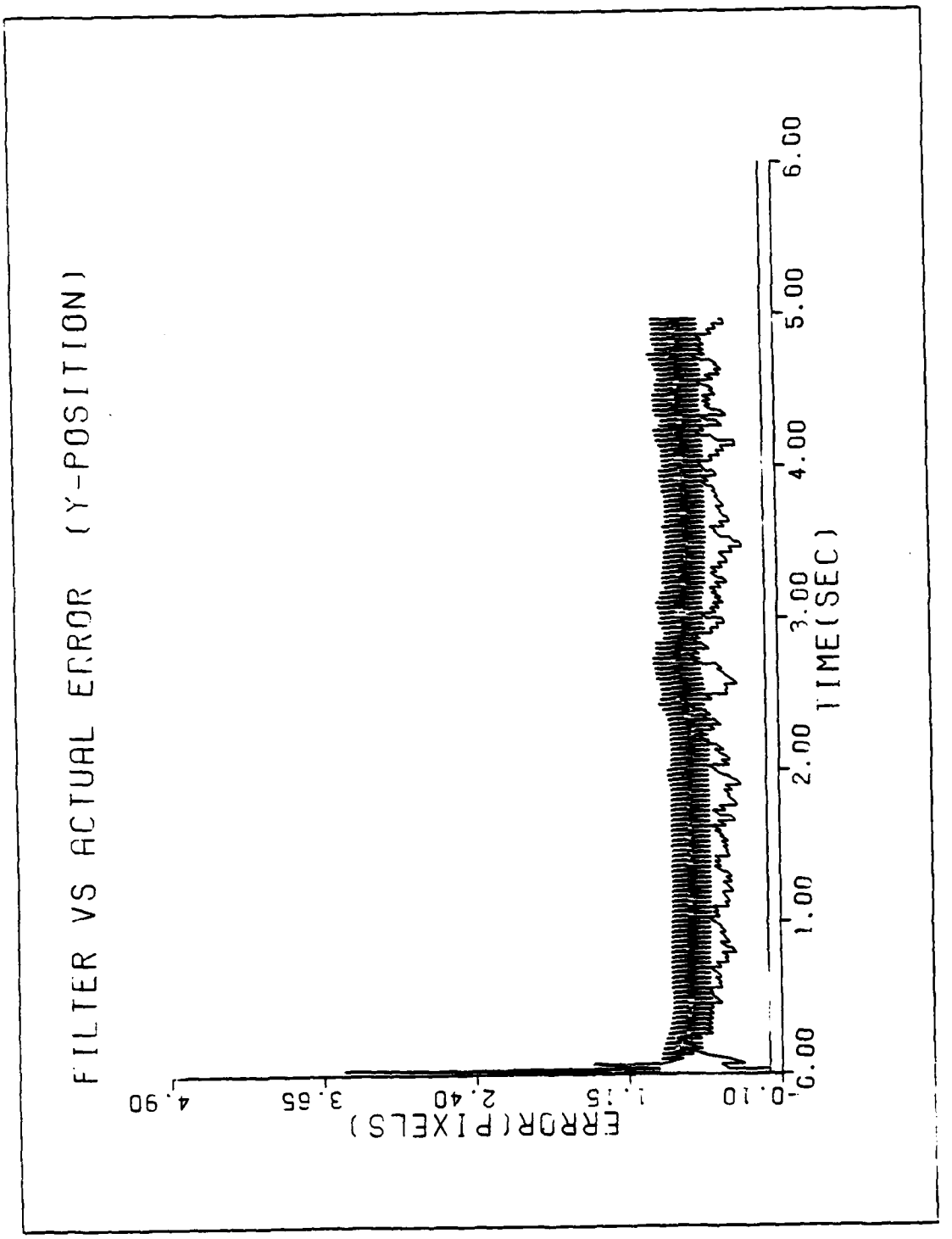


Figure B-3b. Performance Plot for T2G2:M-BAY

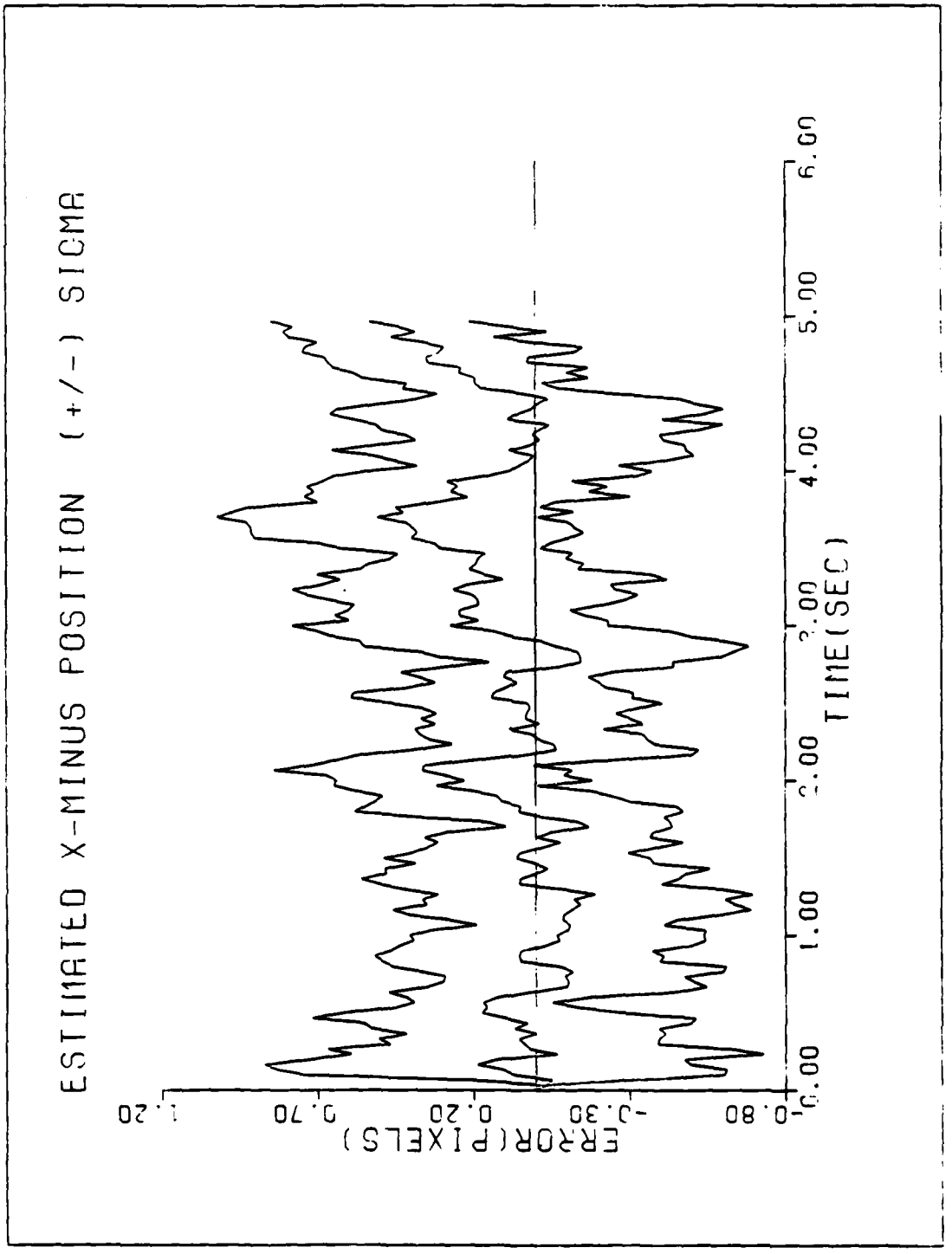


Figure B-3c. Performance Plot for T2G2M1-BAY

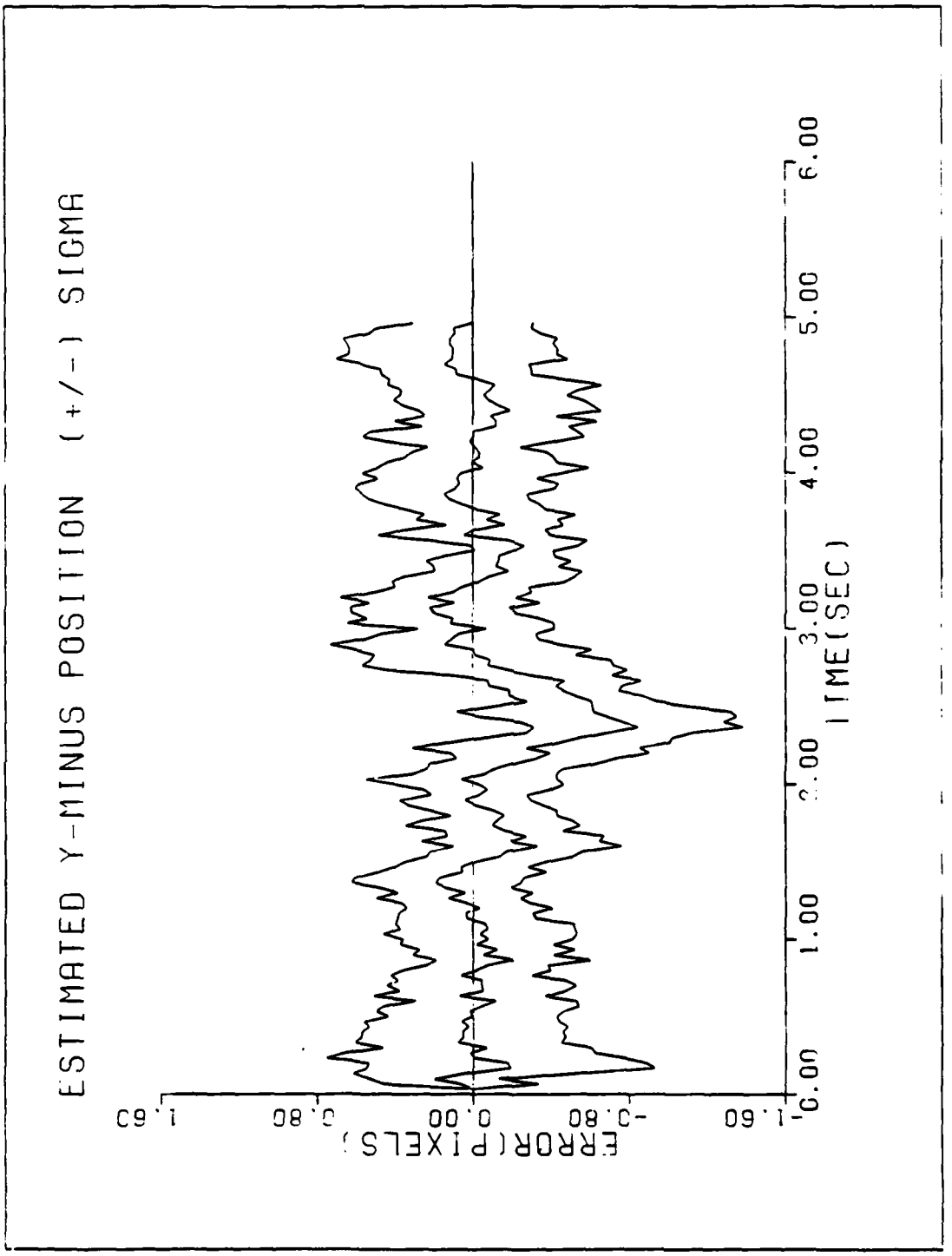


Figure B-3d. Performance Plot for TOGGLE-BAY

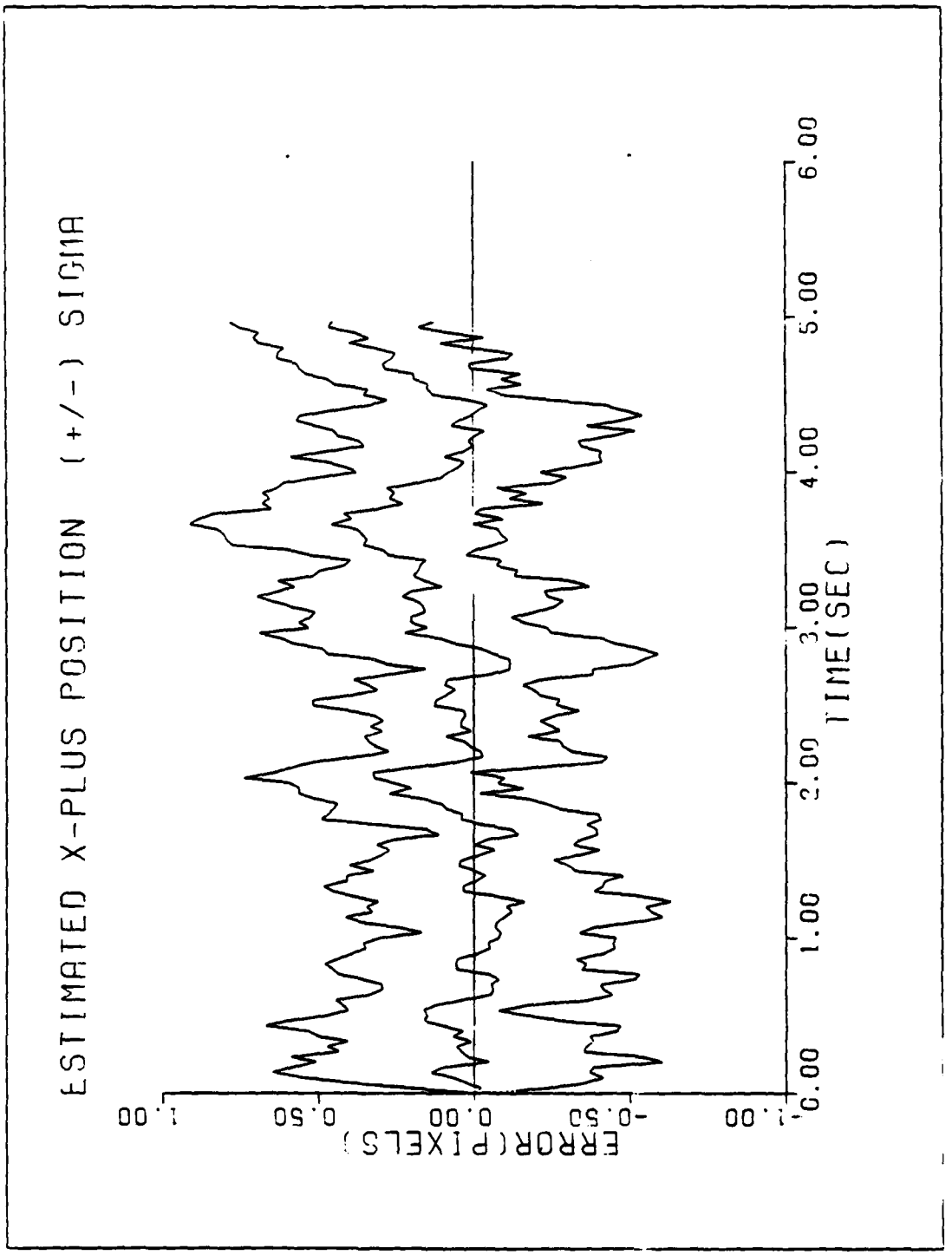


Figure B-3e. Performance Plot for T2G2MM-BAY

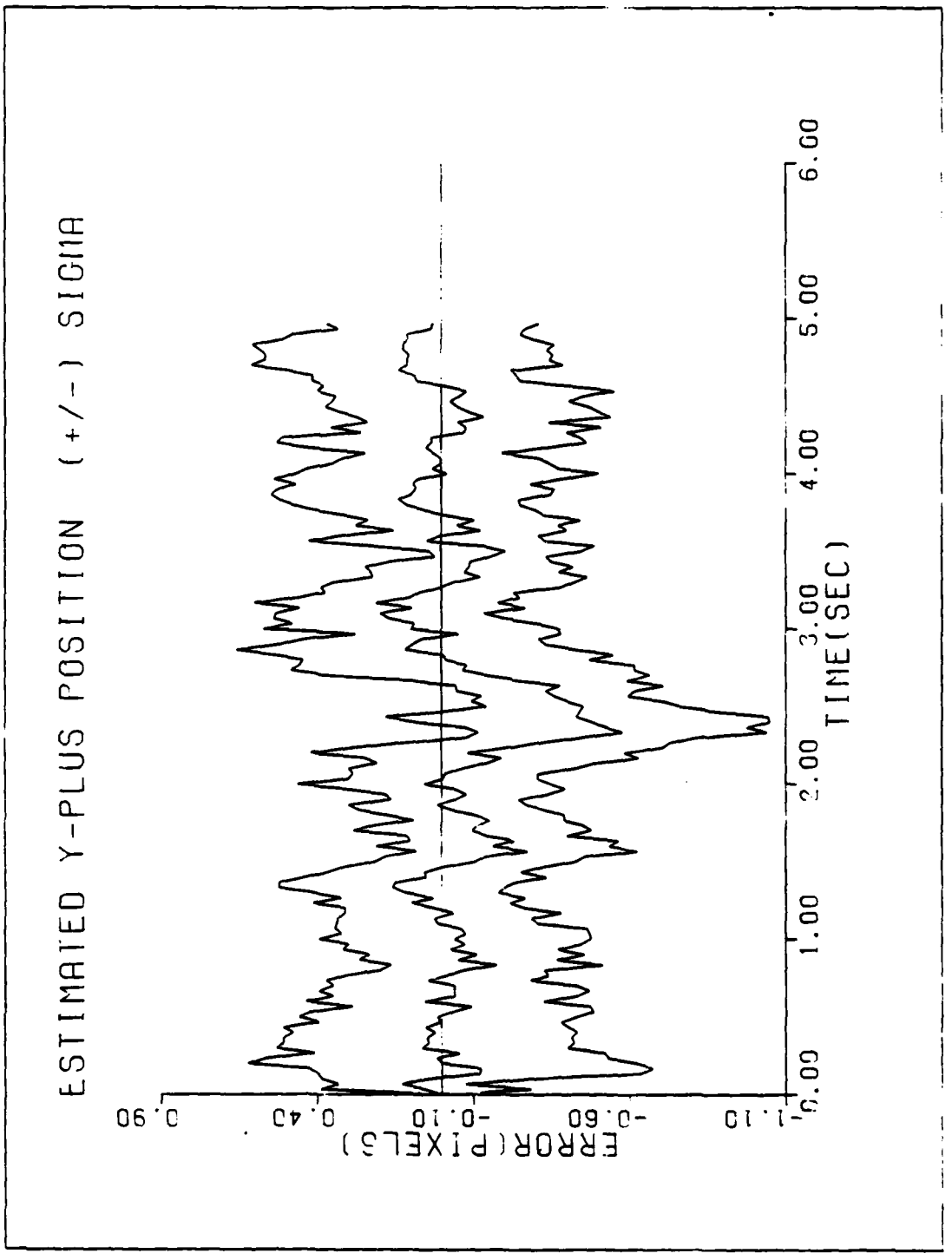


Figure B-3f. Performance Plot for T2G2M1-BAY

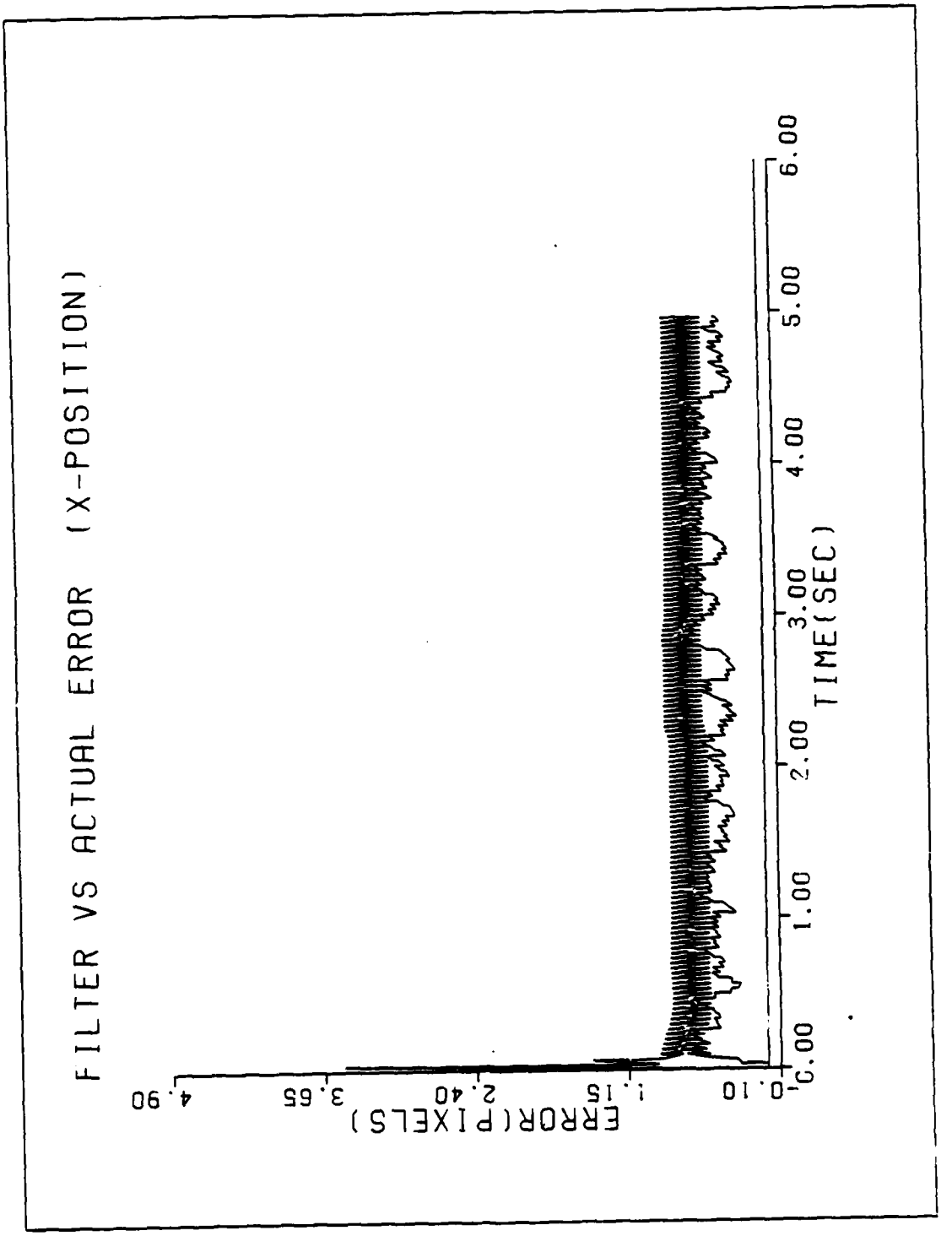


Figure B-4a. Performance Plot for TCG10MM-BAY

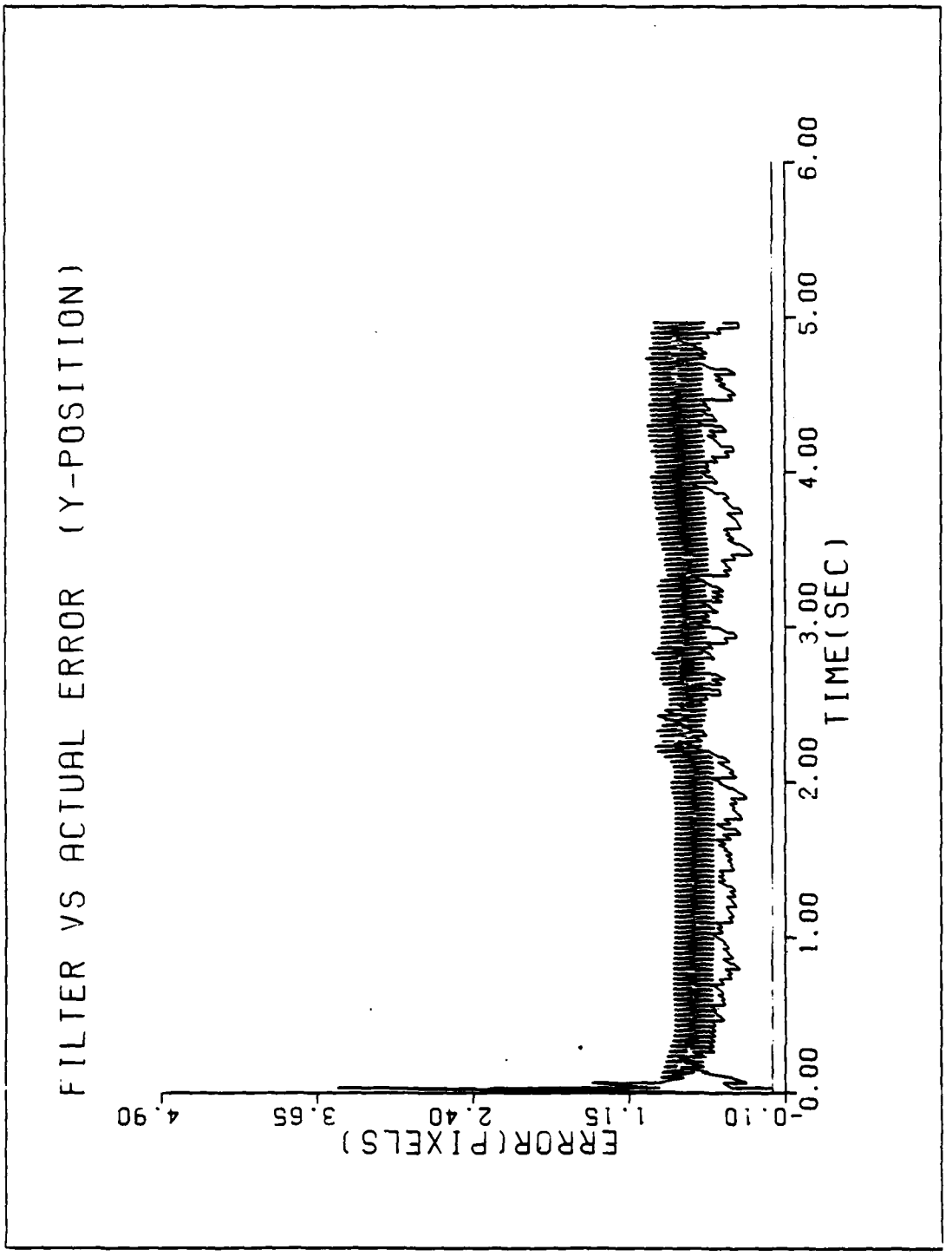


Figure B-4b. Performance Plot for T2G10MM-BAY

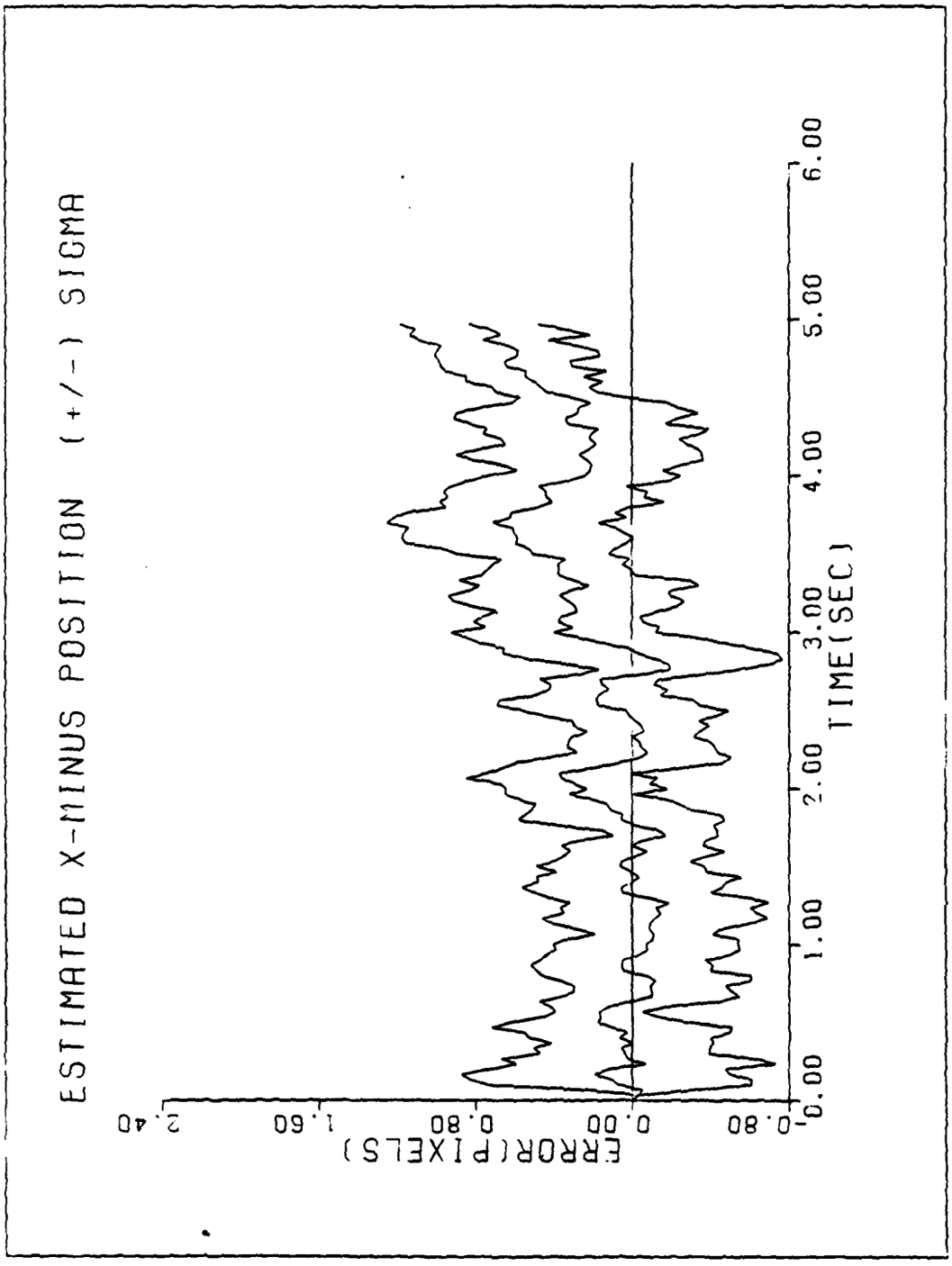


Figure B-4c. Performance Plot for T2G10MM-BAY

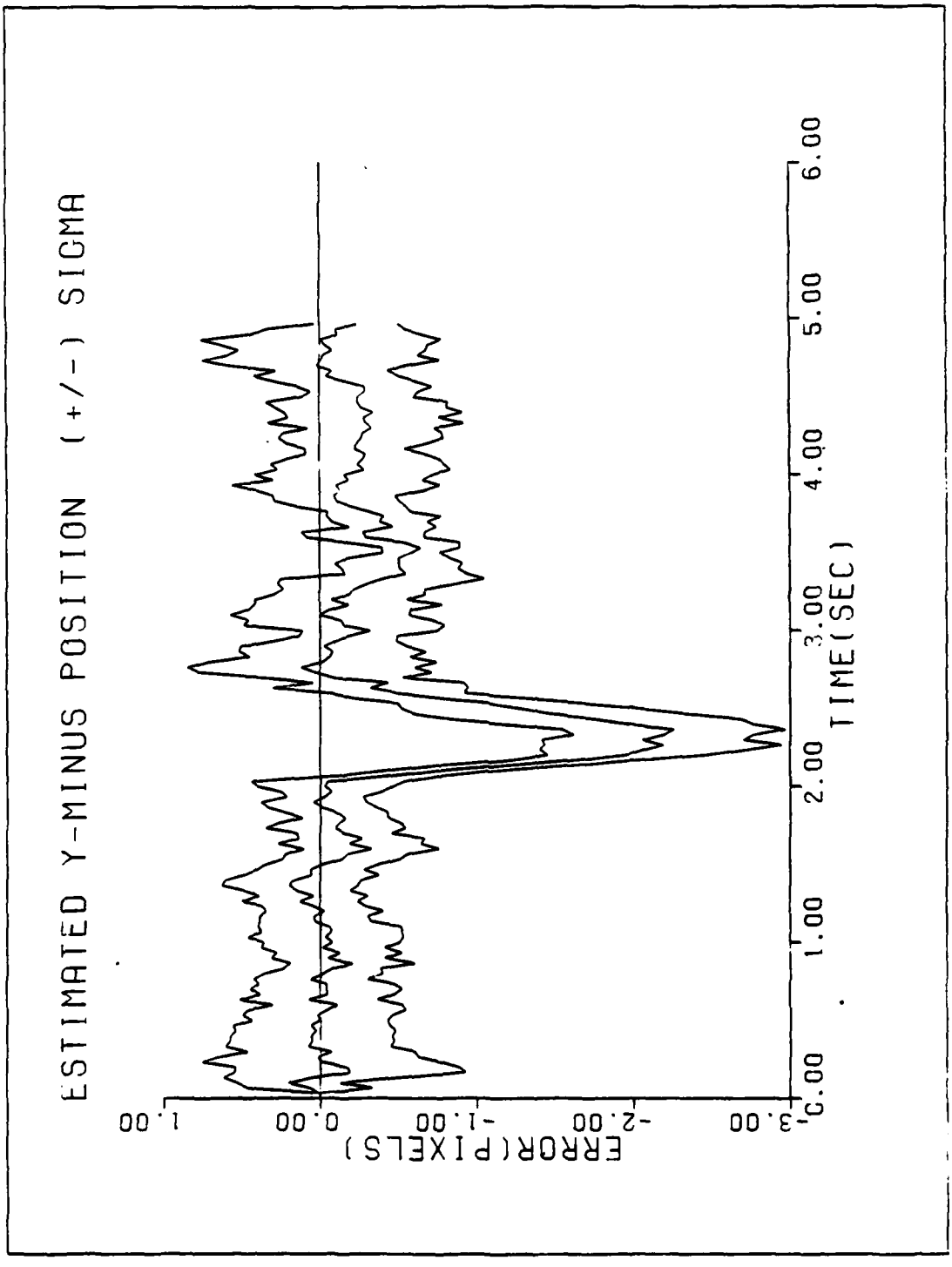


Figure B-4d. Performance Plot for TCG10MI-BAY

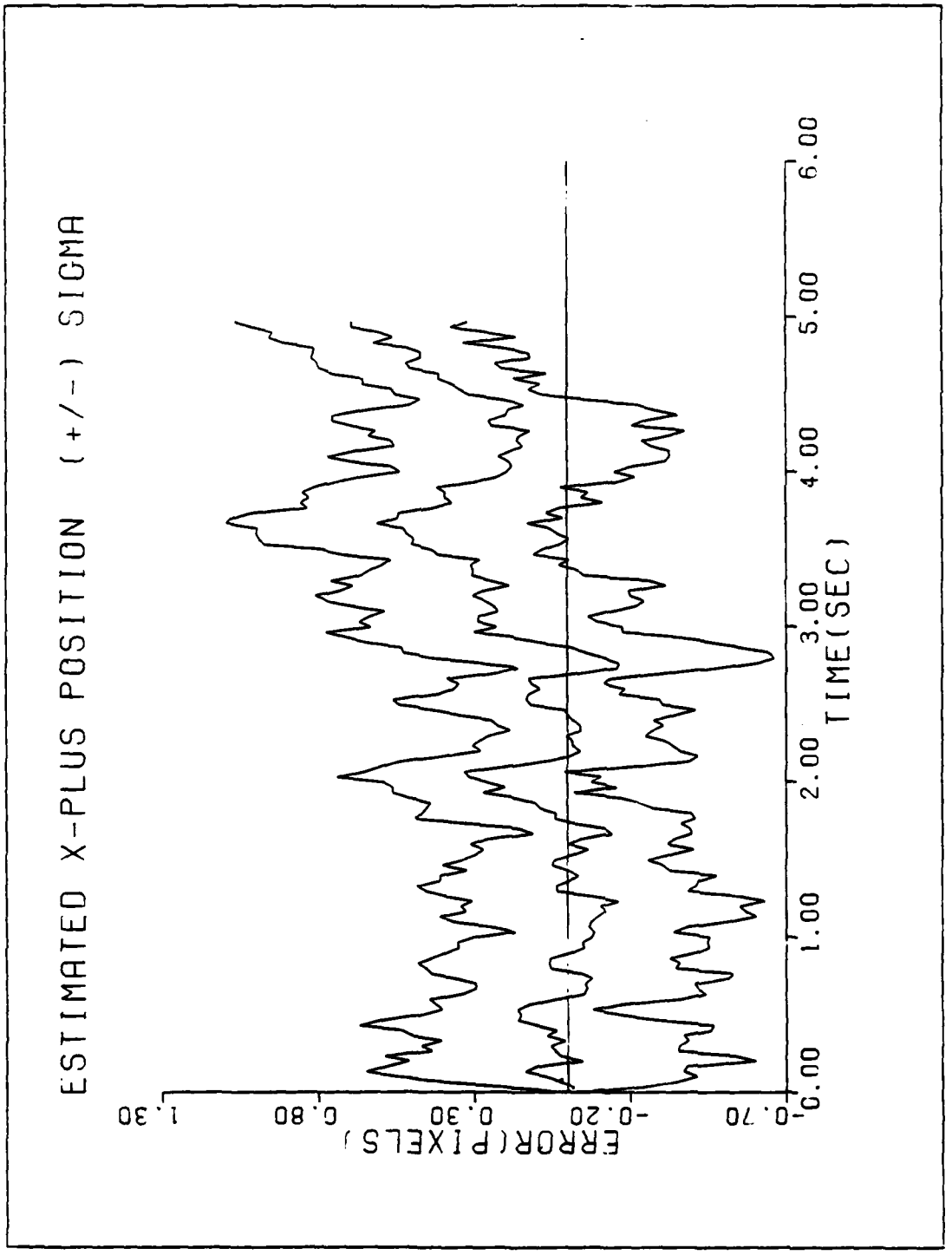


Figure B-4e. Performance Plot for T2G10MM-BAY

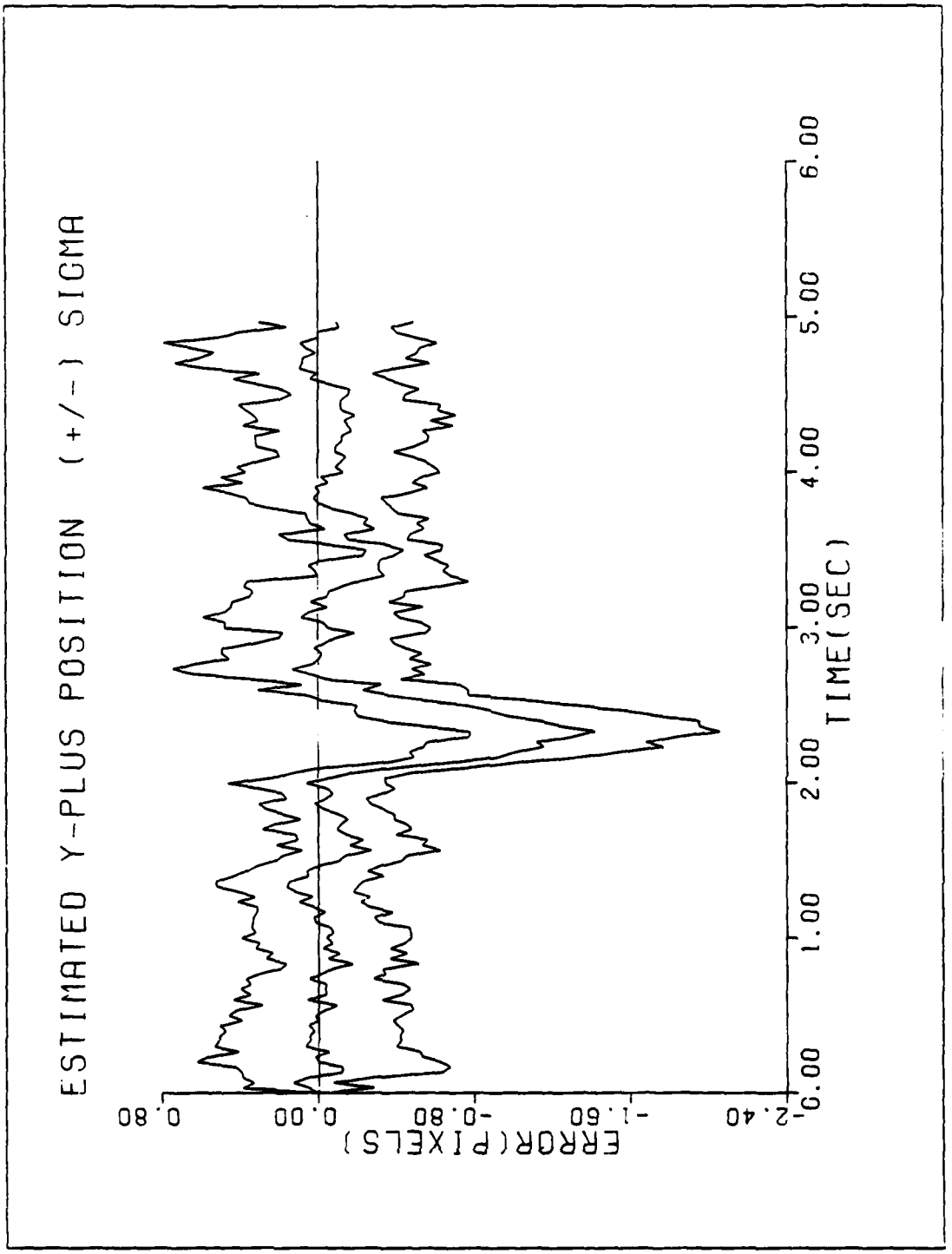


Figure B-4f. Performance Plot for [illegible]

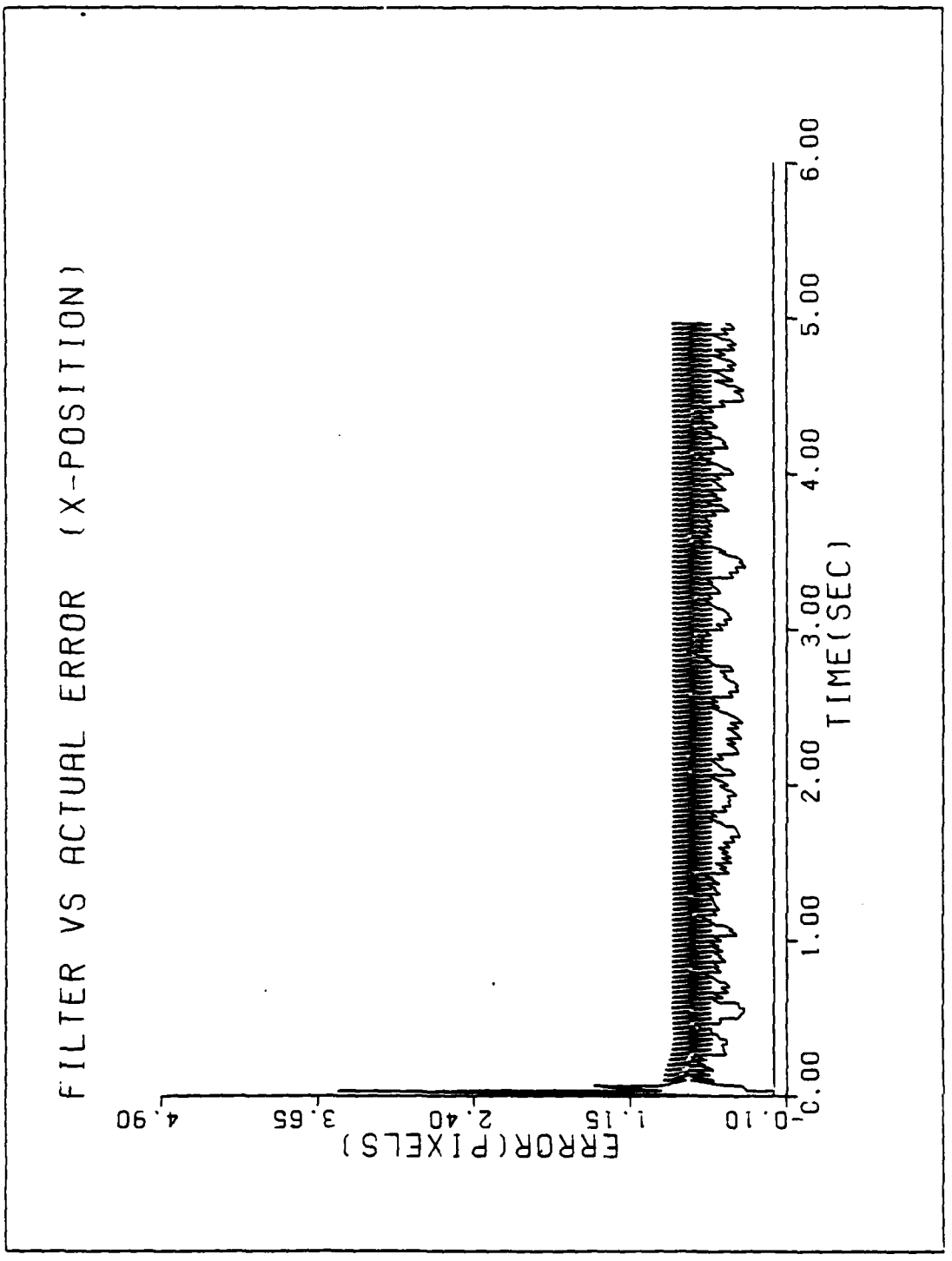


Figure B-5a. Performance Plot for T2G10F3-BAY

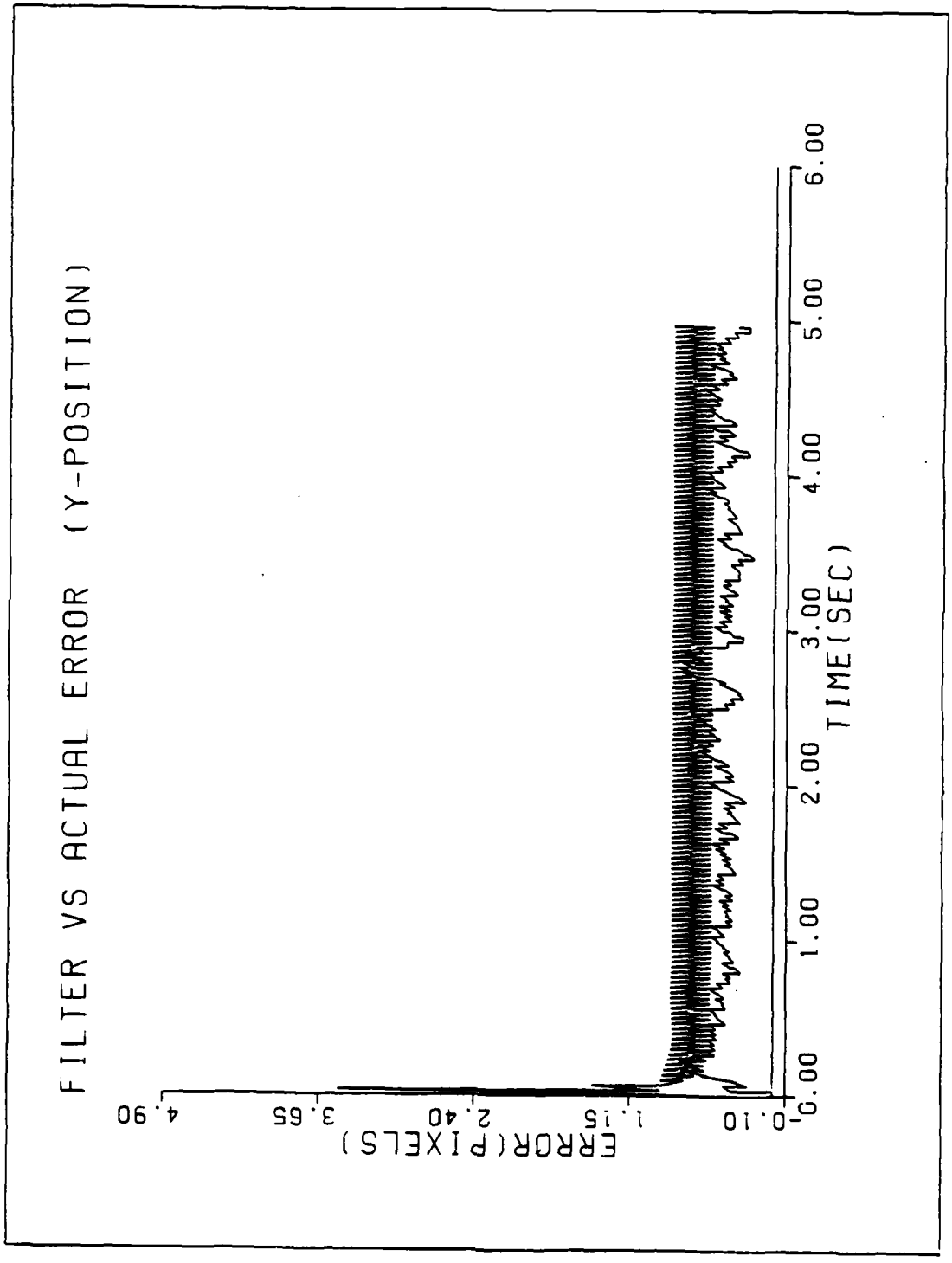


Figure B-5b. Performance Plot for T2G1DF3-BAY

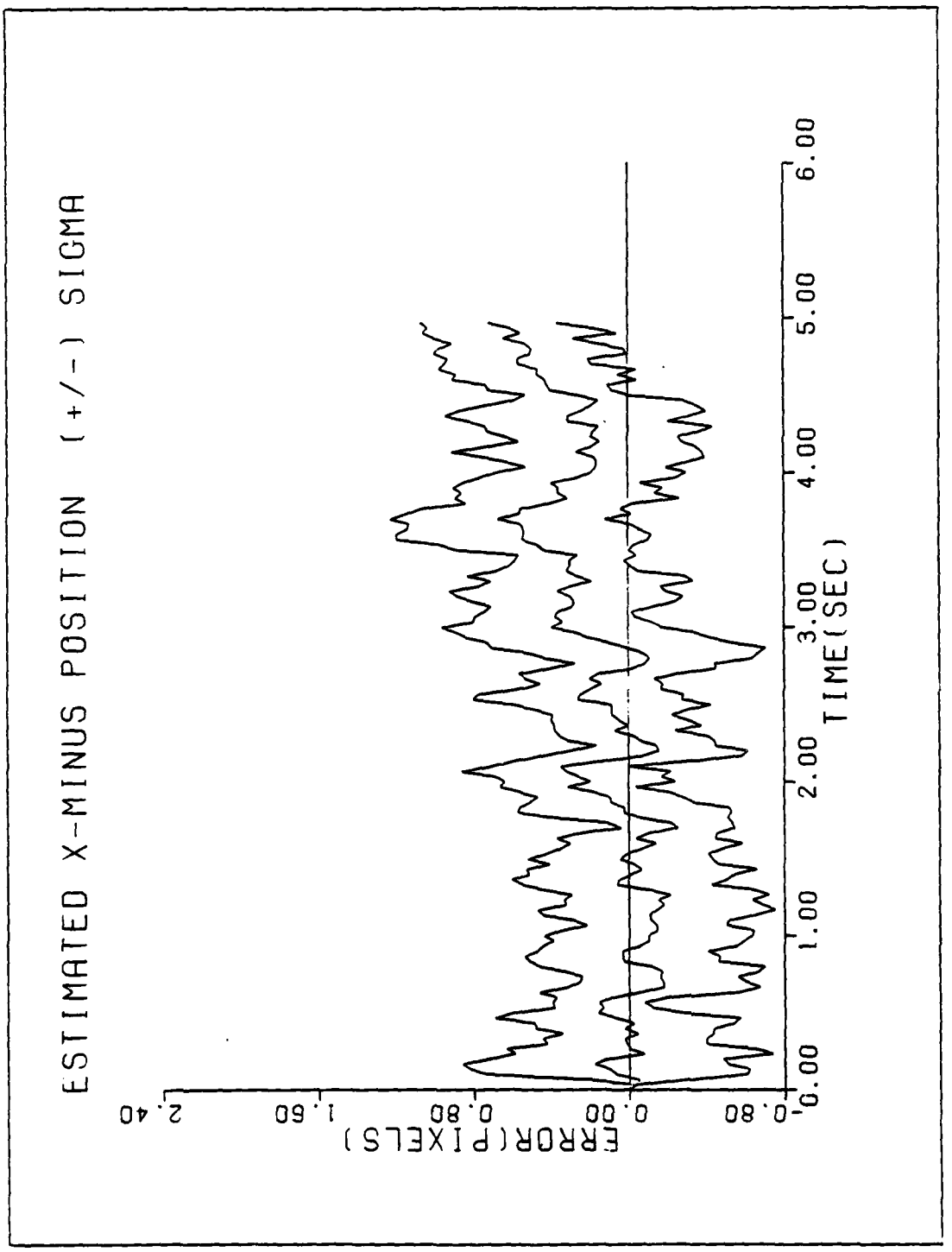


Figure B-5c. Performance Plot for T3G10F3-BAY

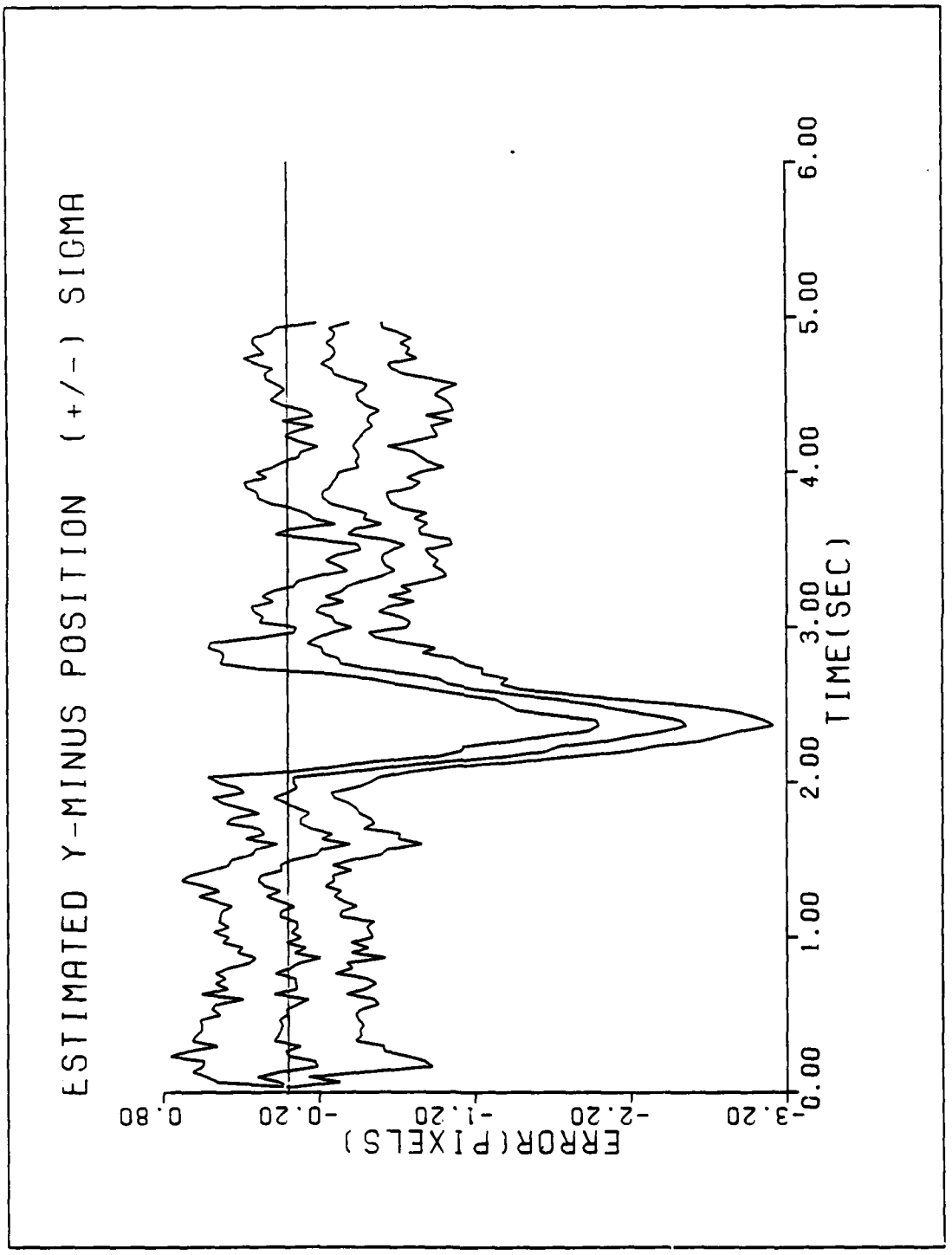


Figure B-5d. Performance Plot for T2G10F3-LAY

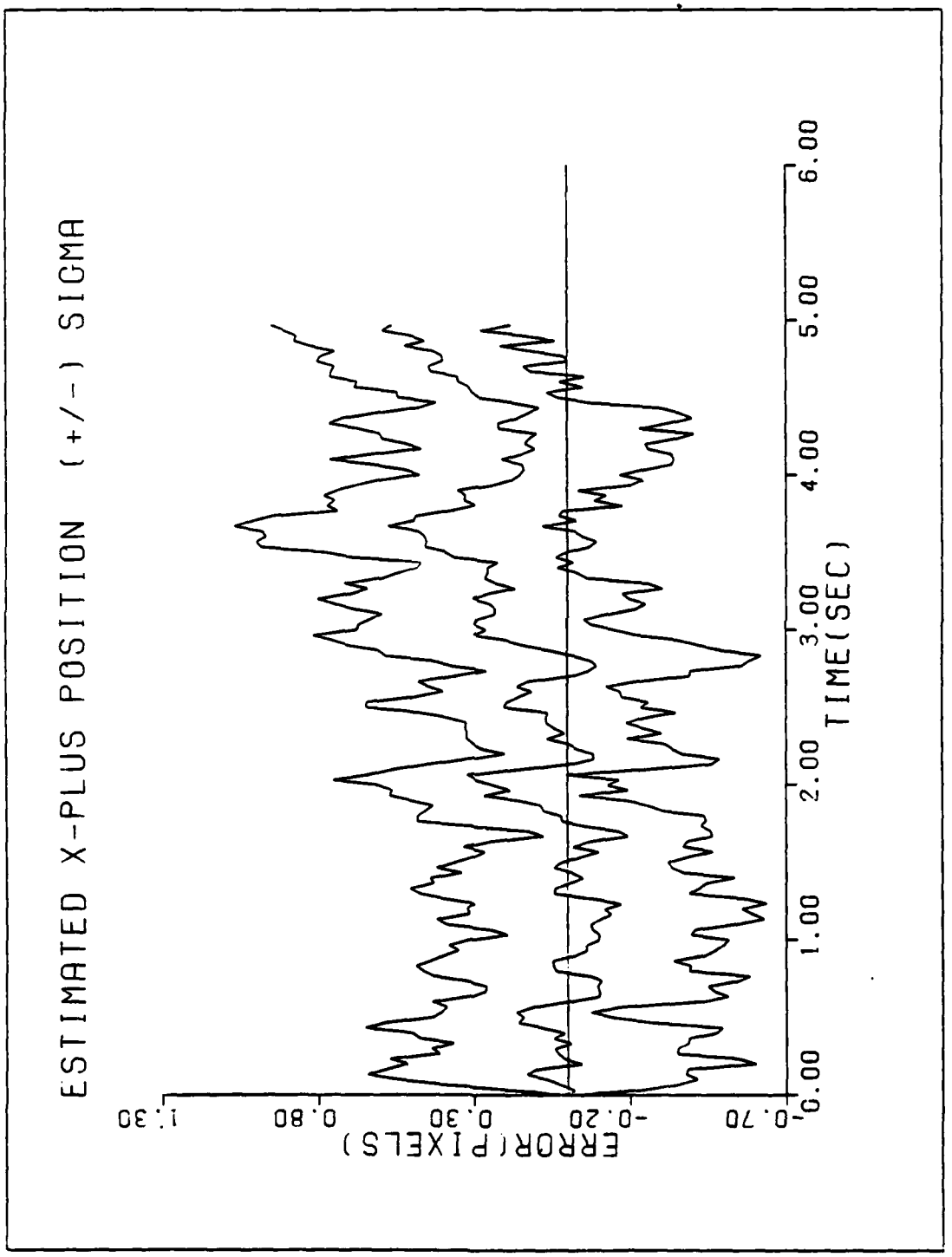


Figure B-5e. Performance Plot for T2G10F3-BAY

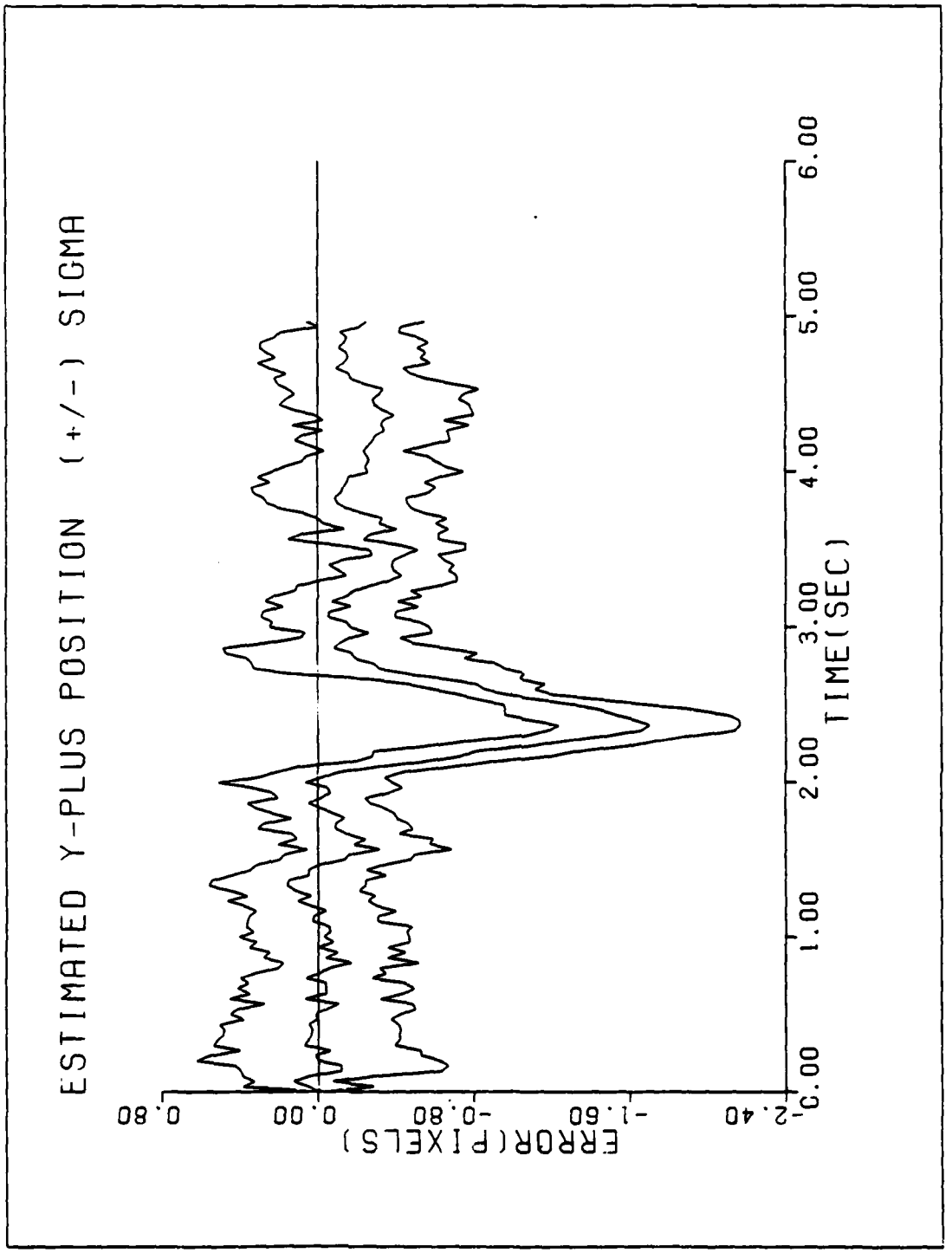


Figure B-5f. Performance Plot for T2G10F3-BAY

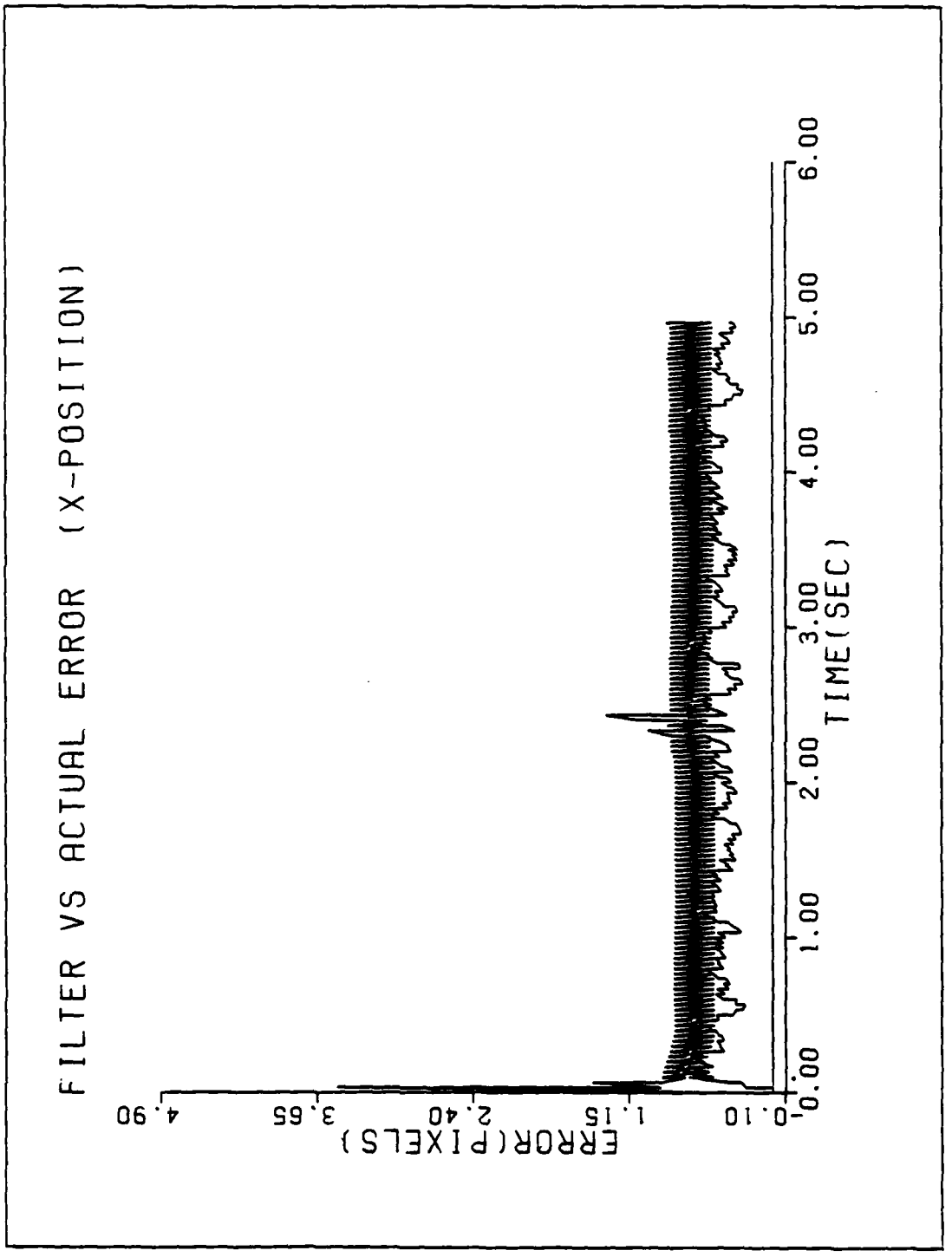


Figure B-6a. Performance Plot for TDGCOMM-BAY

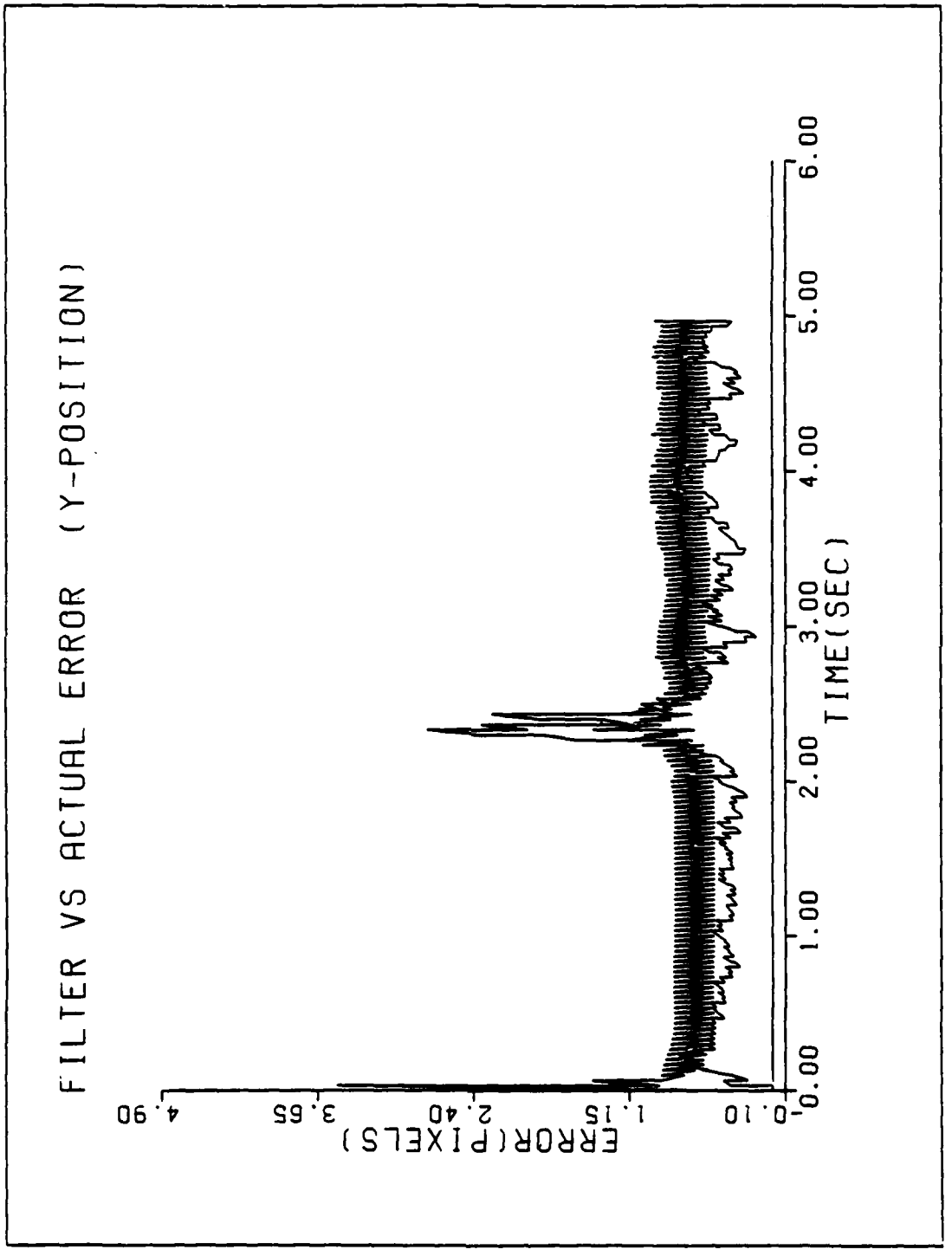


Figure B-6b. Performance Plot for T2G20III-BAY

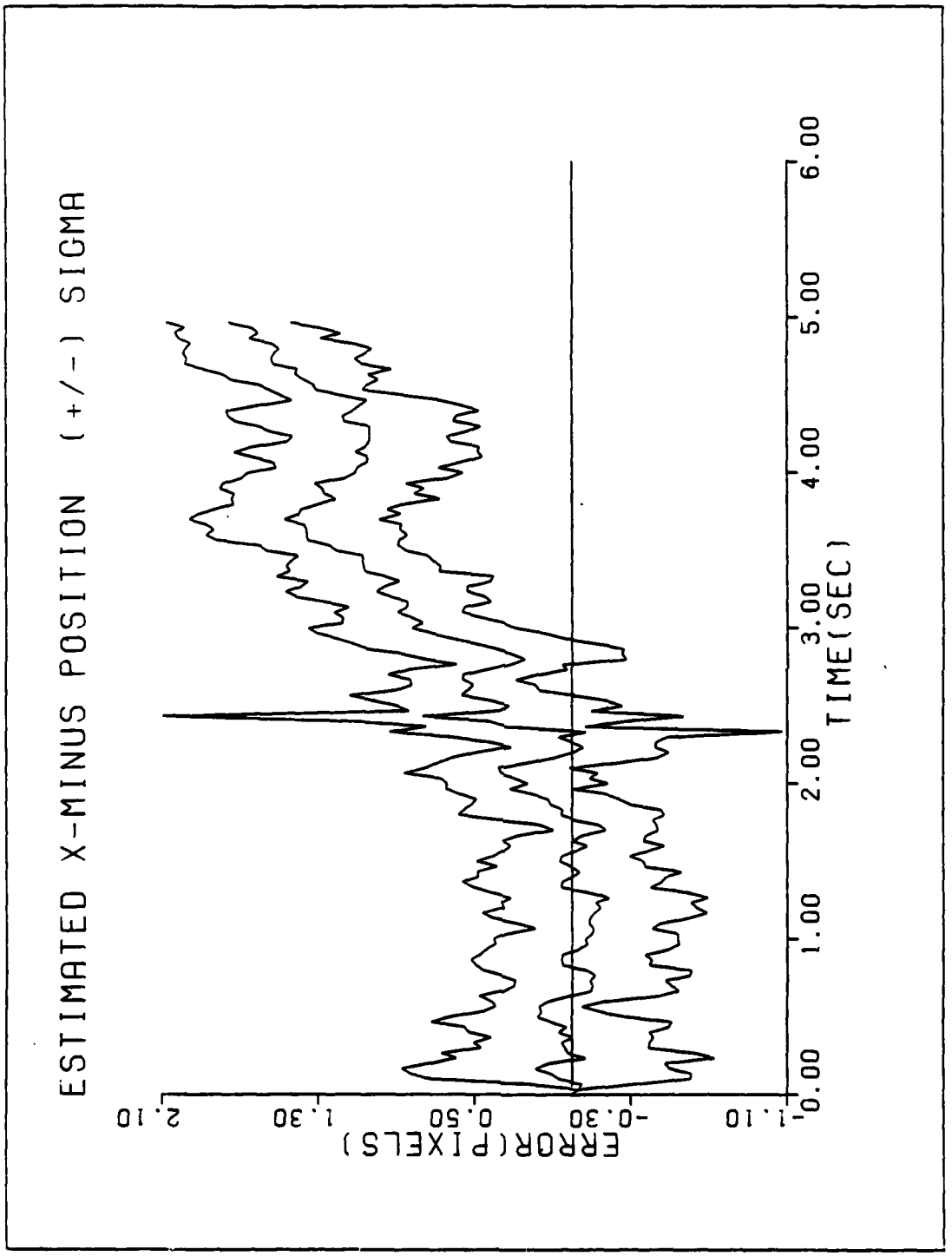


Figure B-6c. Performance Plot for TRGCOMM-BAY

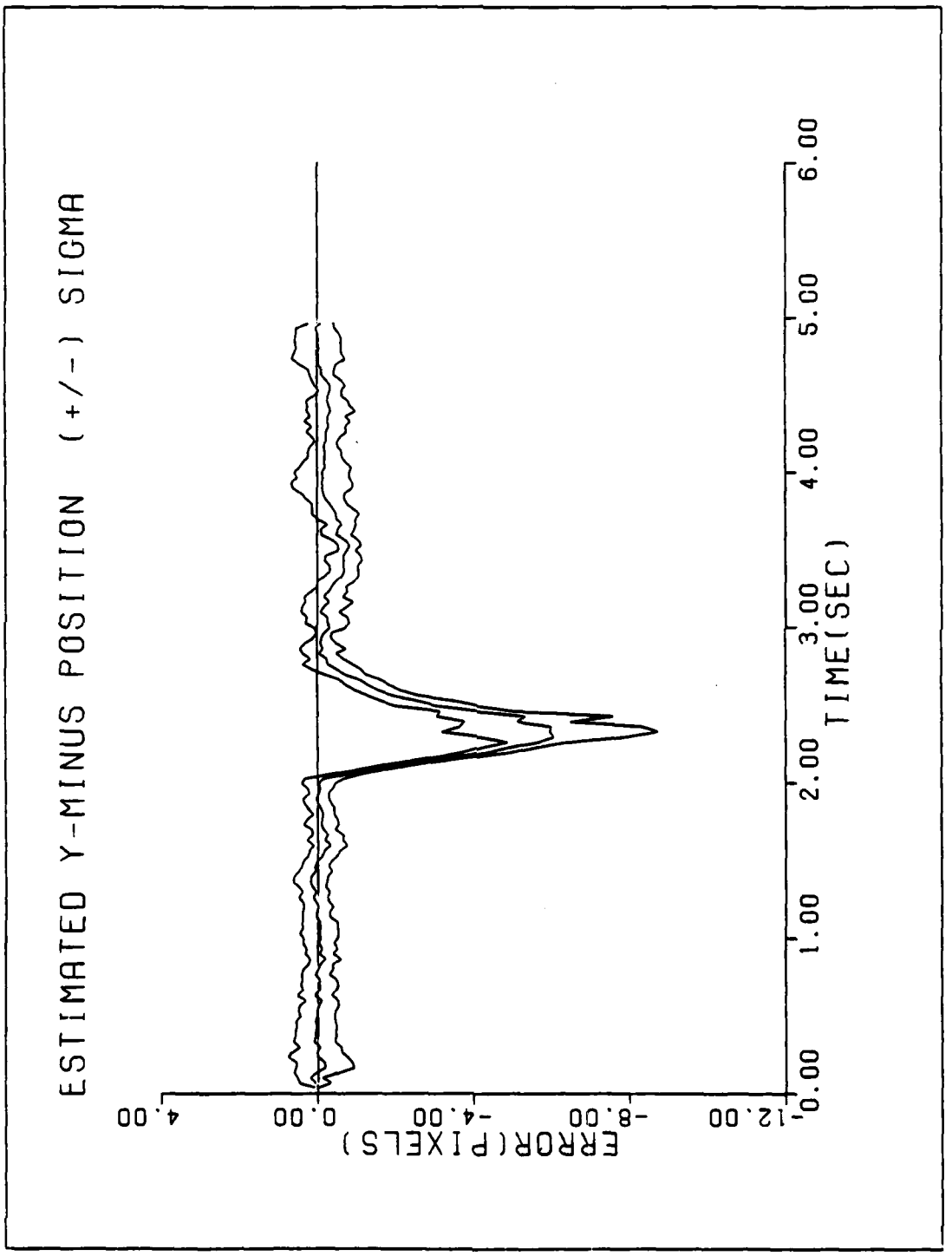


Figure B-6d. Performance Plot for T2G2OMI-BAY

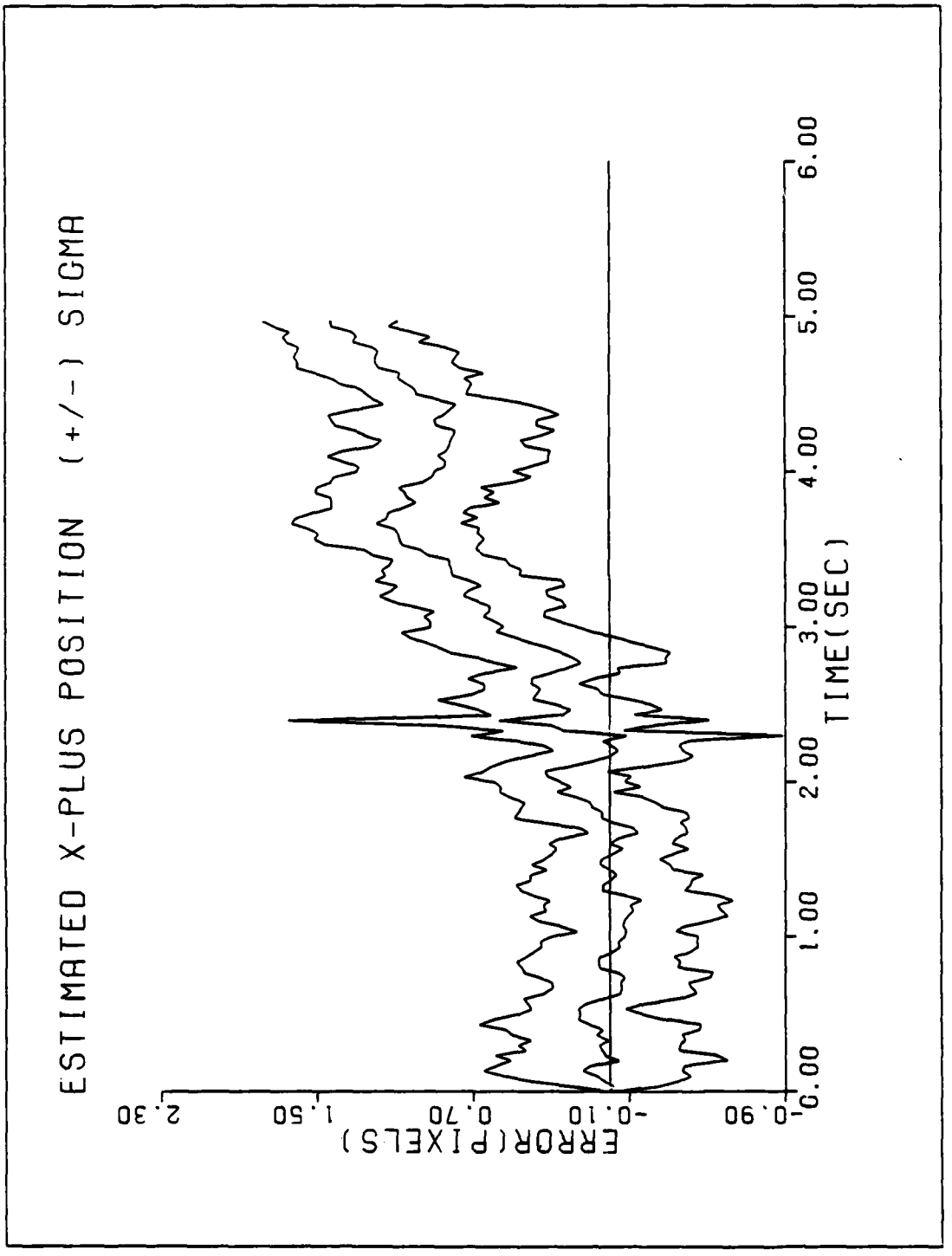


Figure B-6e. Performance Plot for TEGCOMM-BAY

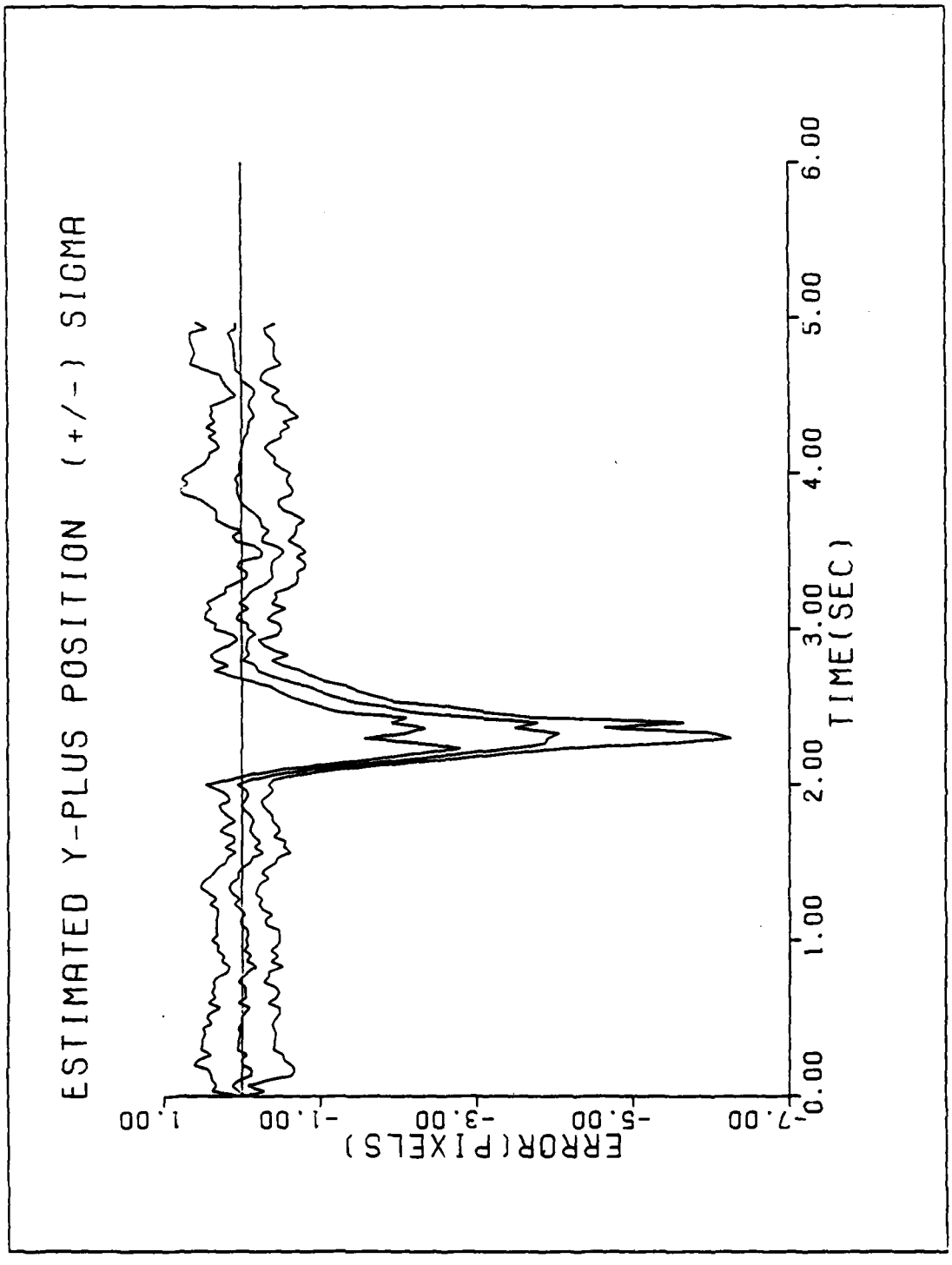


Figure B-6f. Performance Plot for T2C2CMI-BAY

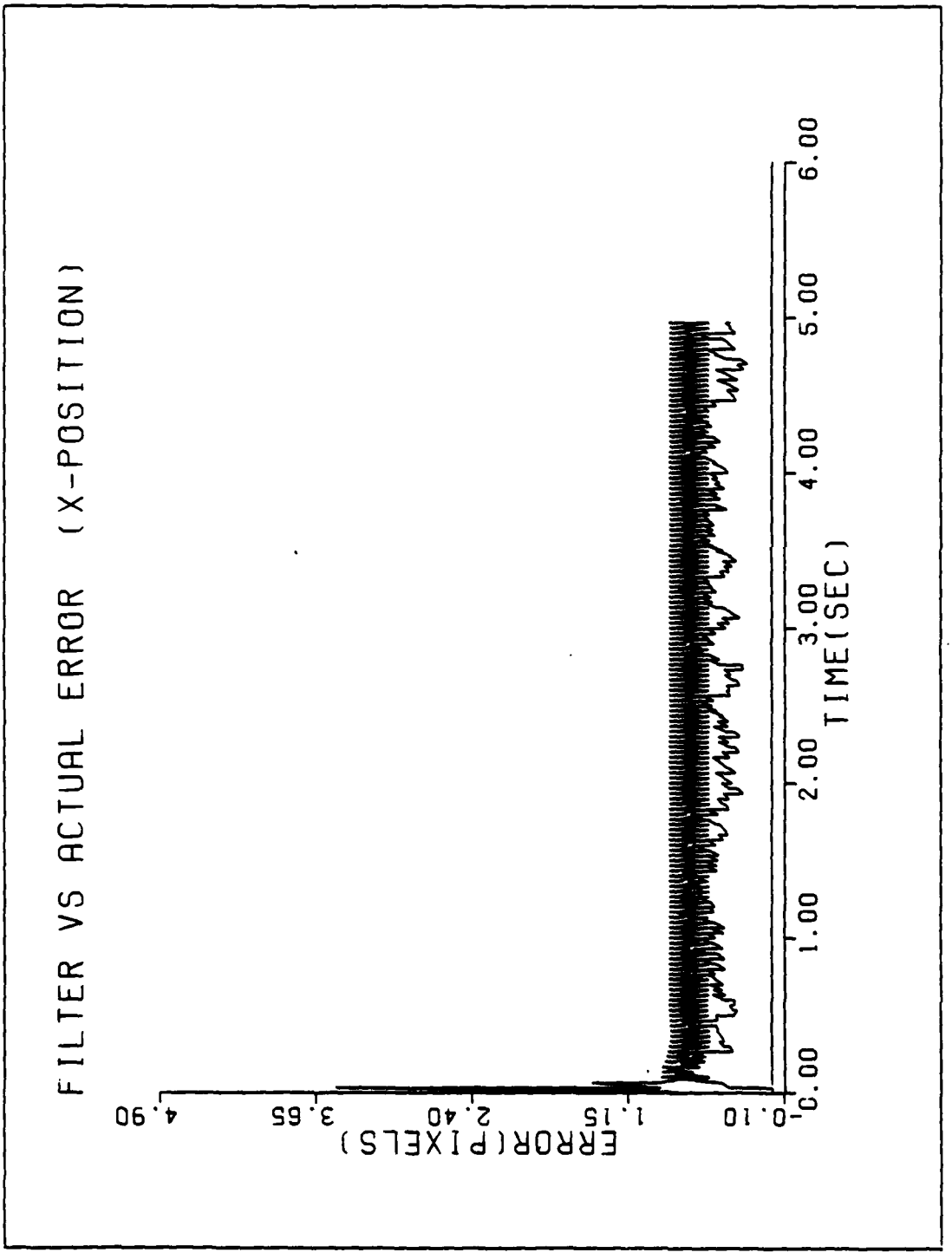


Figure B-7a. Performance Plot for T2G2OF2-BAY

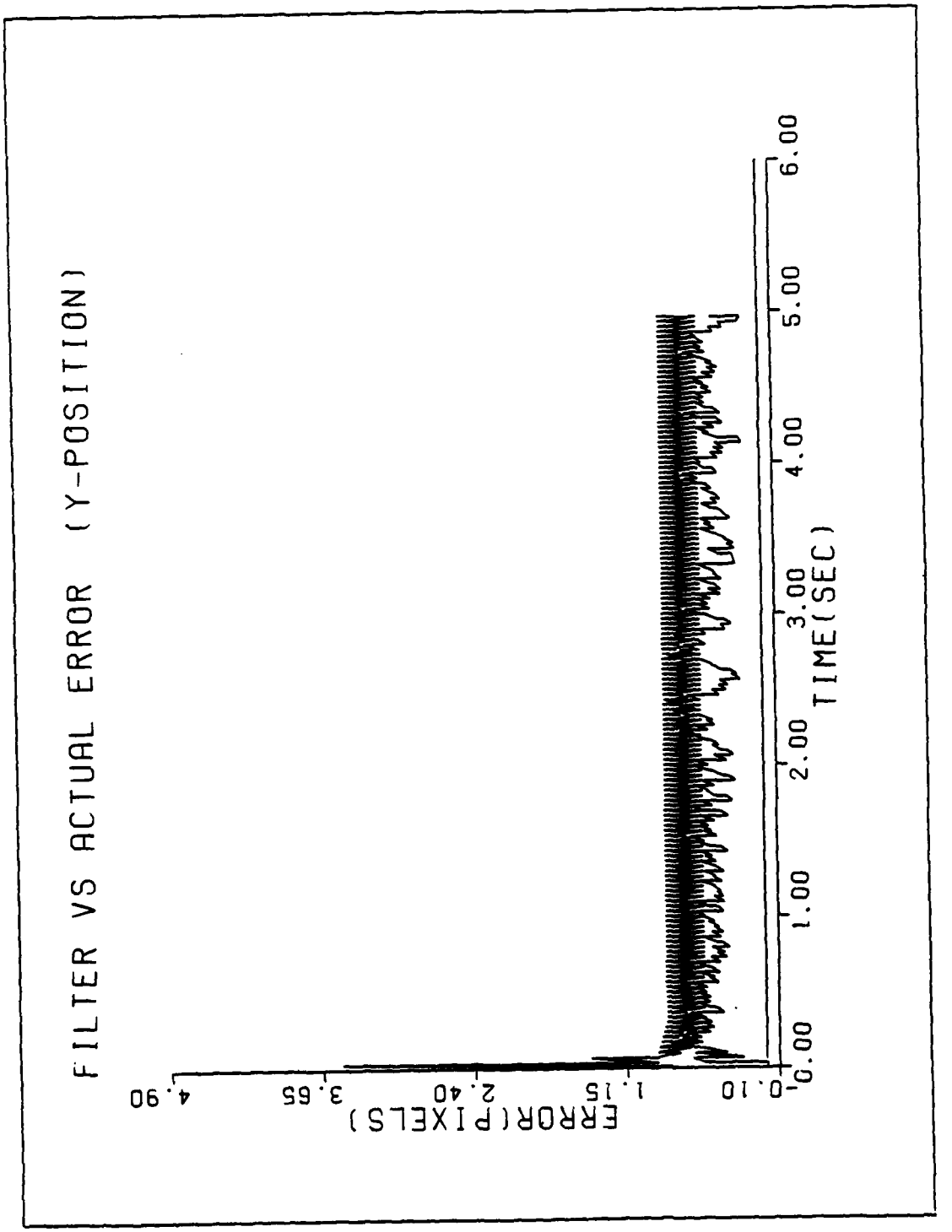


Figure B-7b. Performance Plot for T2G2OF2-BAY

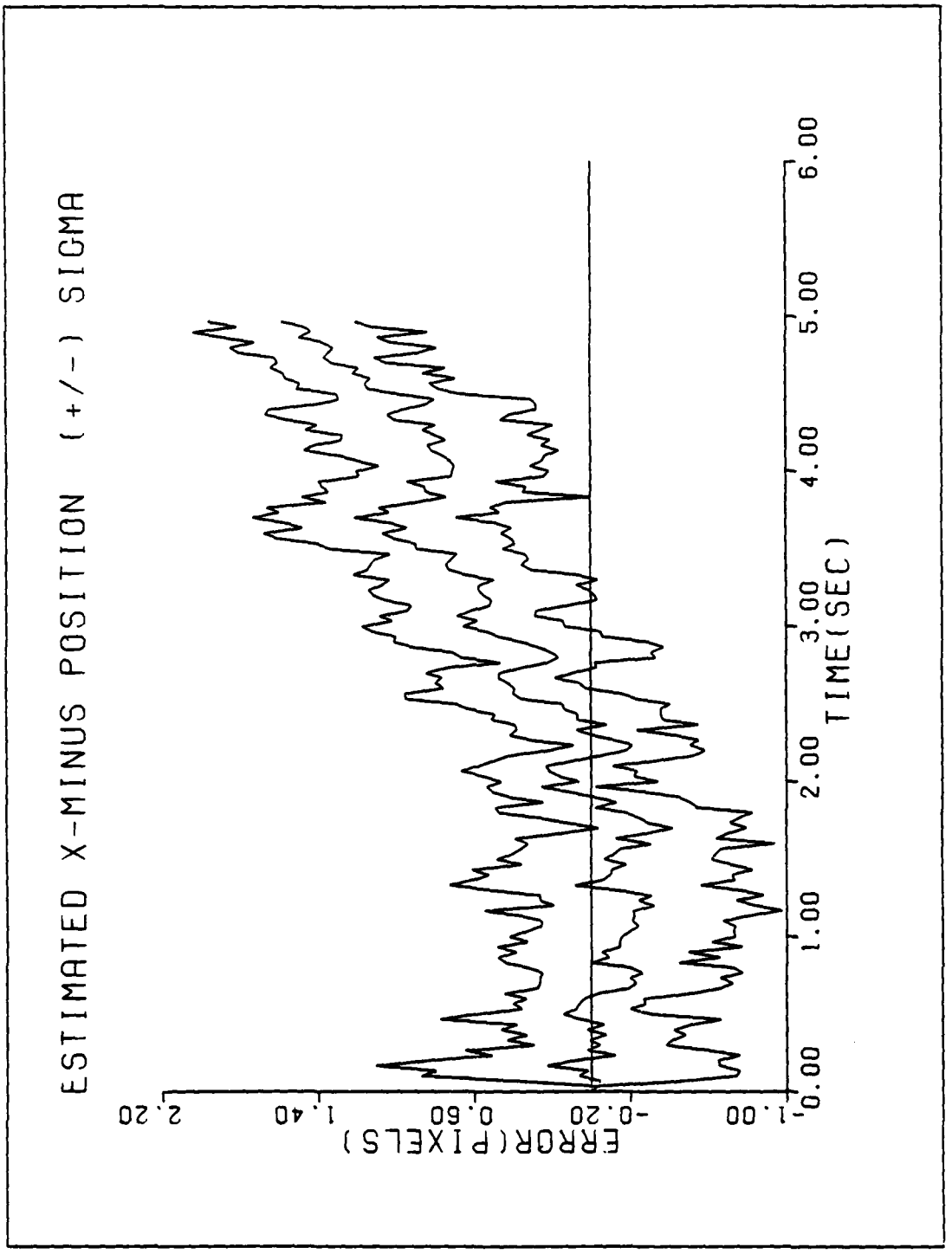


Figure B-7c. Performance Plot for T2GC0F2-BAY

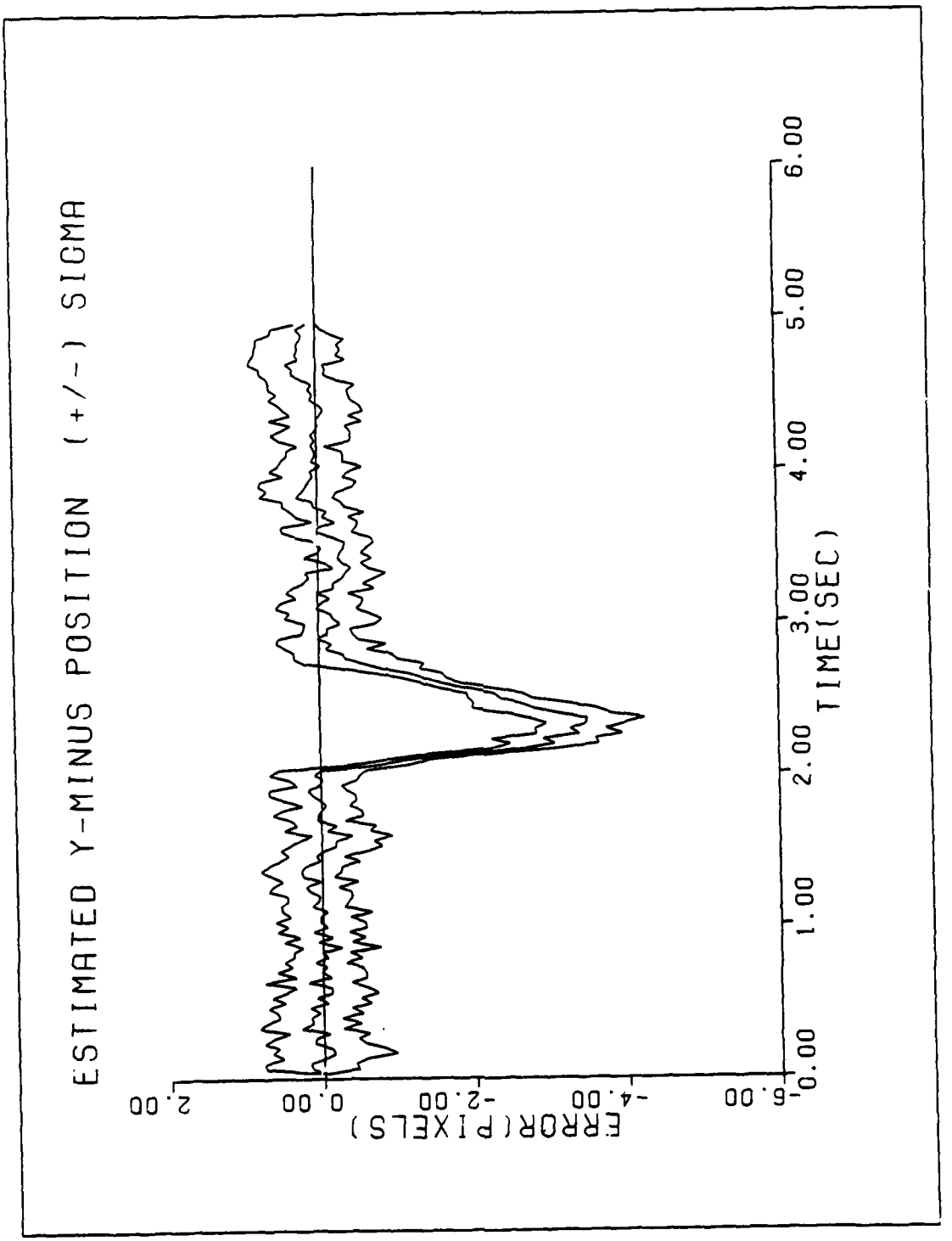
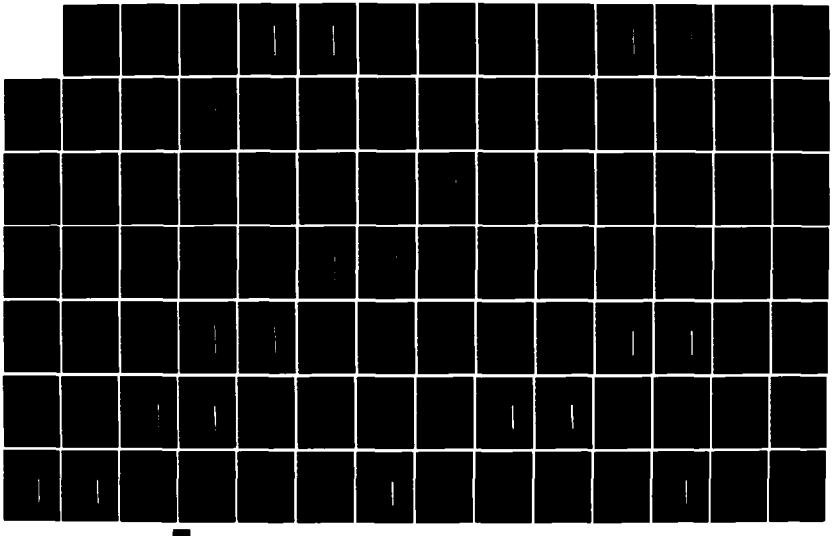
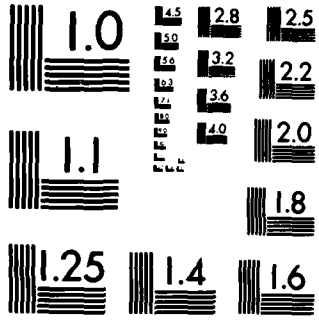


Figure B-7d. Performance Plot for TOGGOYE-BAY

AD-A155 466 BAYESIAN VS MAP MULTIPLE MODEL ADAPTIVE ESTIMATION FOR FIELD OF VIEW EXPA..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. P A LOVING 3/4

UNCLASSIFIED MAR 85 AFIT/GE/EE/85M-1 F/G 12/1 NL





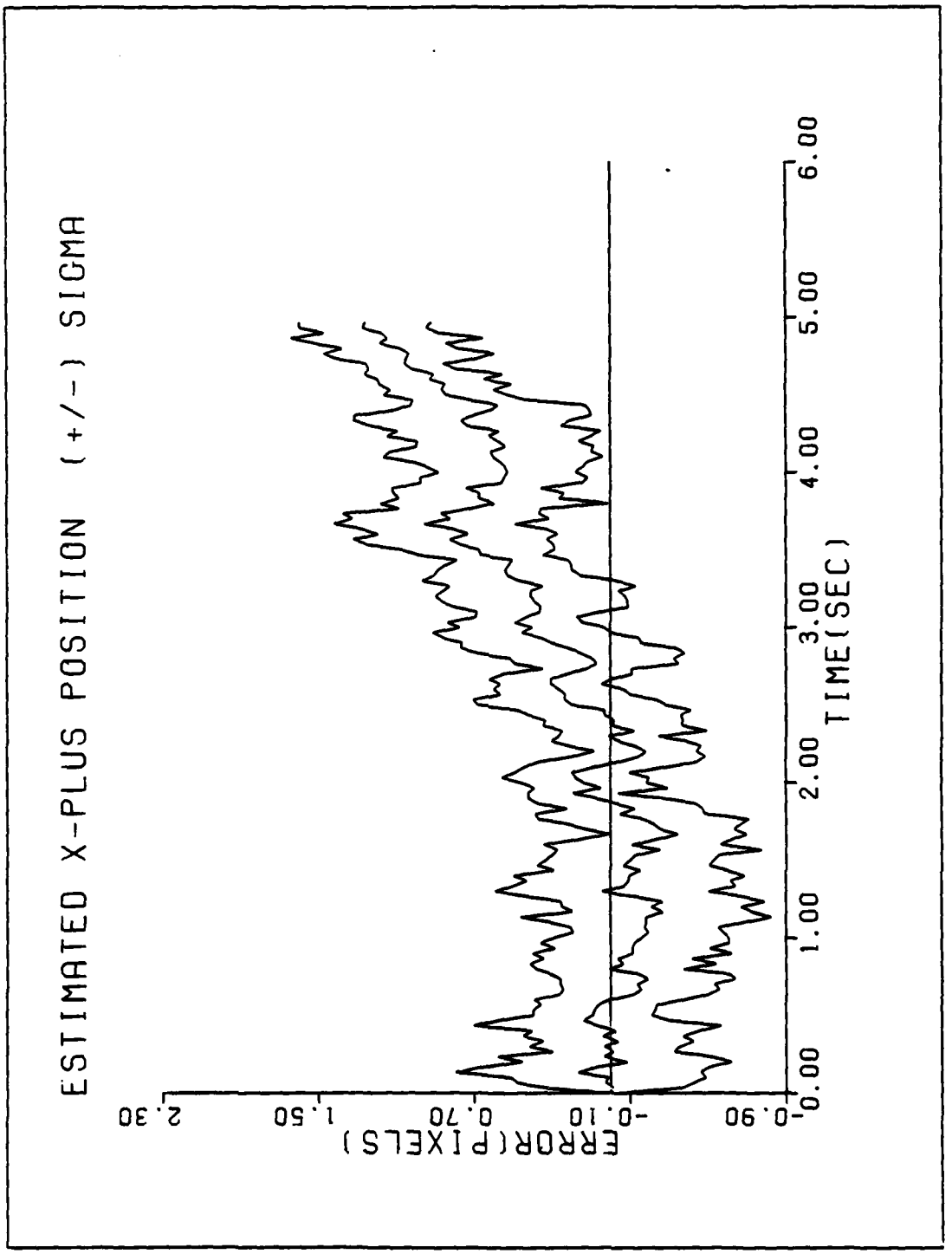


Figure B-7e. Performance Plot for TRGCOFC-BAY

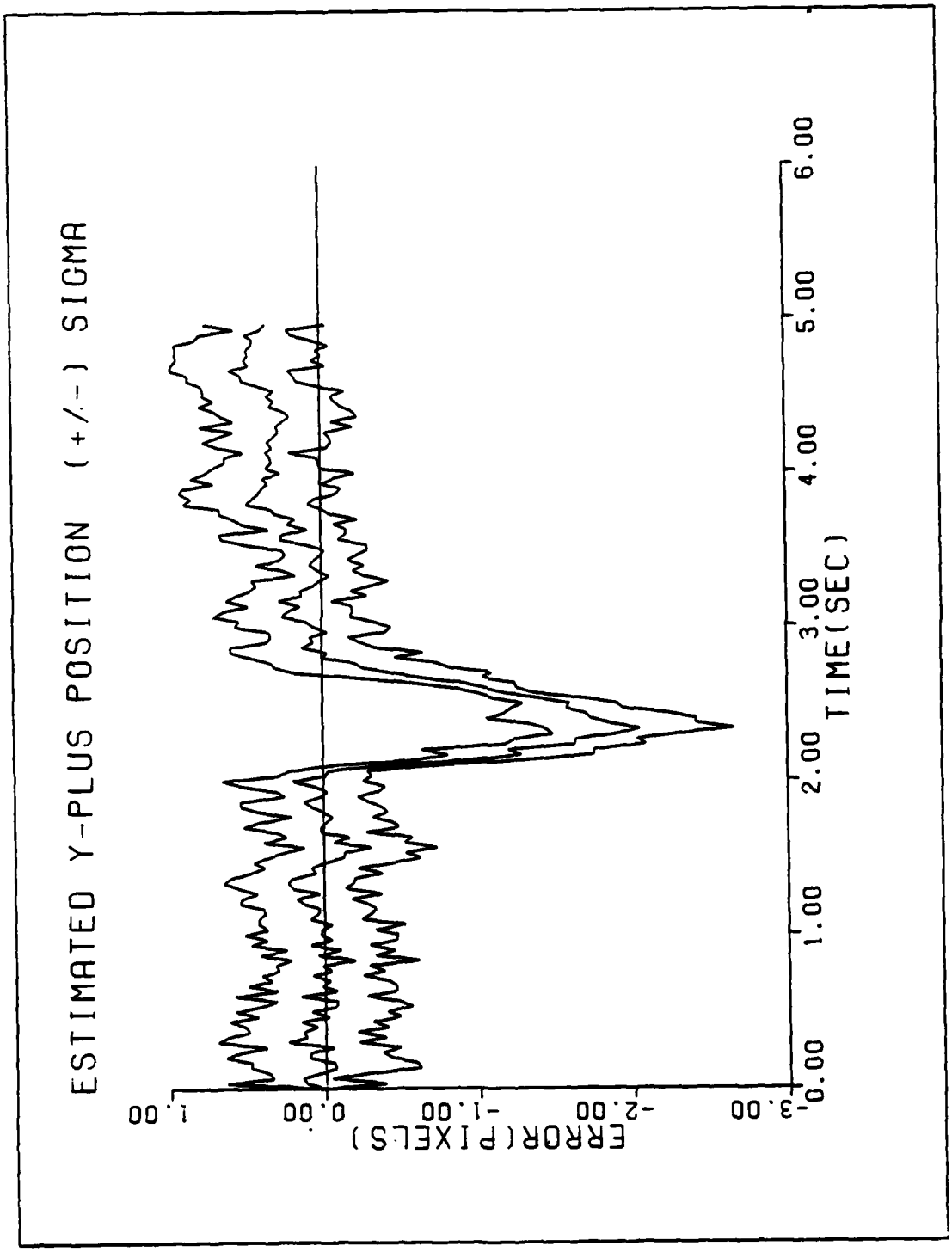


Figure B-7f. Performance Plot for T2G20F2-BAY

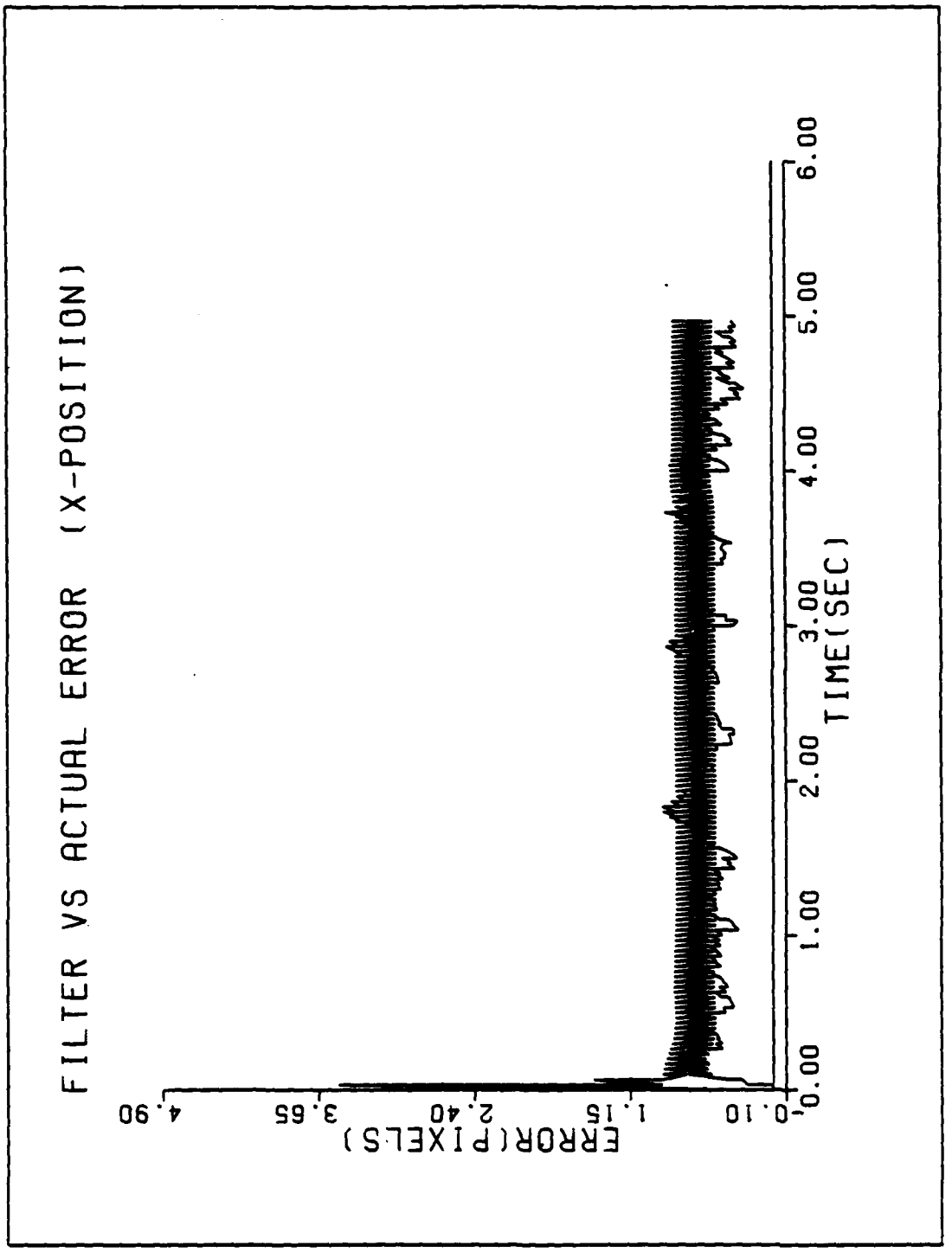


Figure B-8a. Performance Plot for T3G2MI-BAY

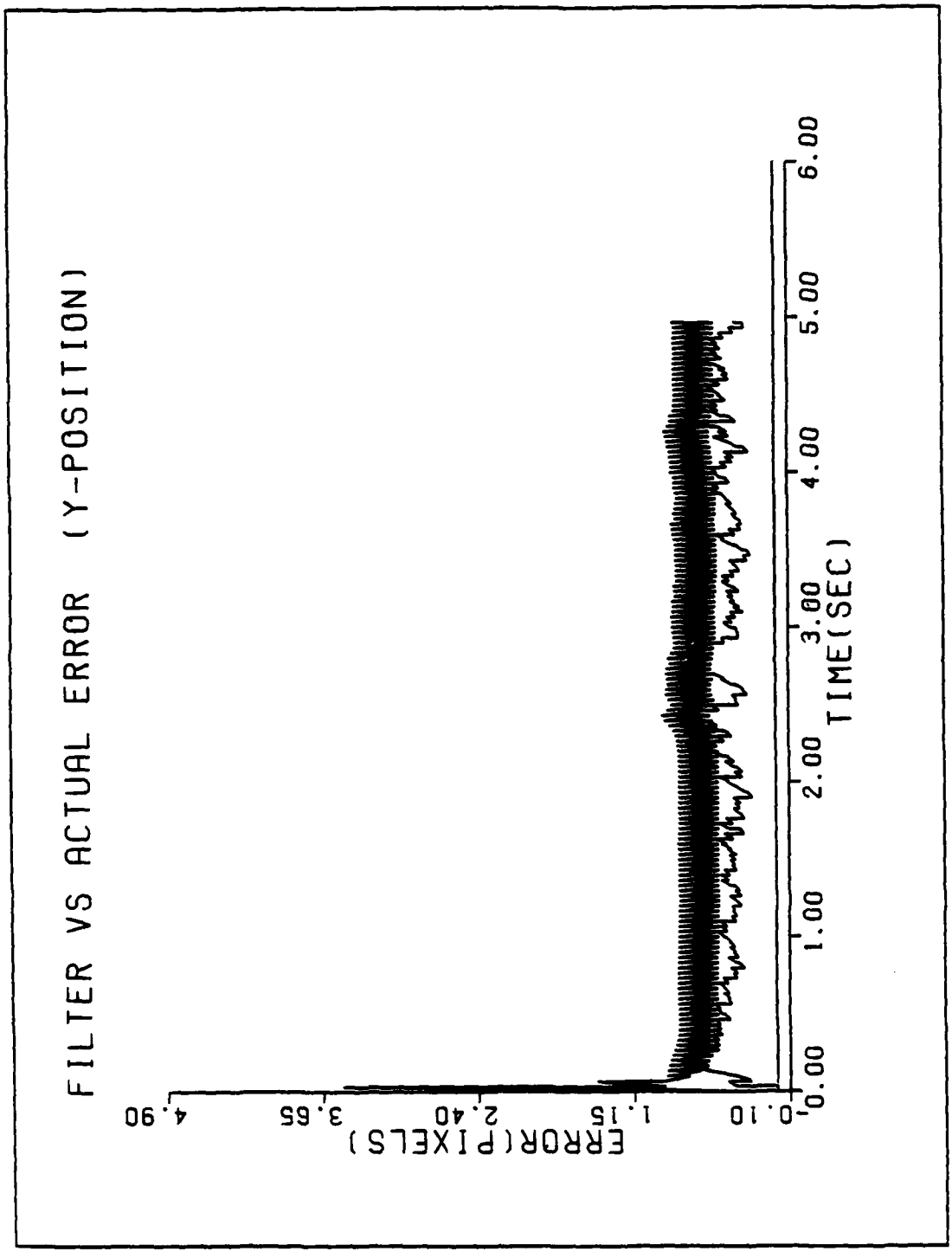


Figure B-8b. Performance Plot for TOG2M-BAY

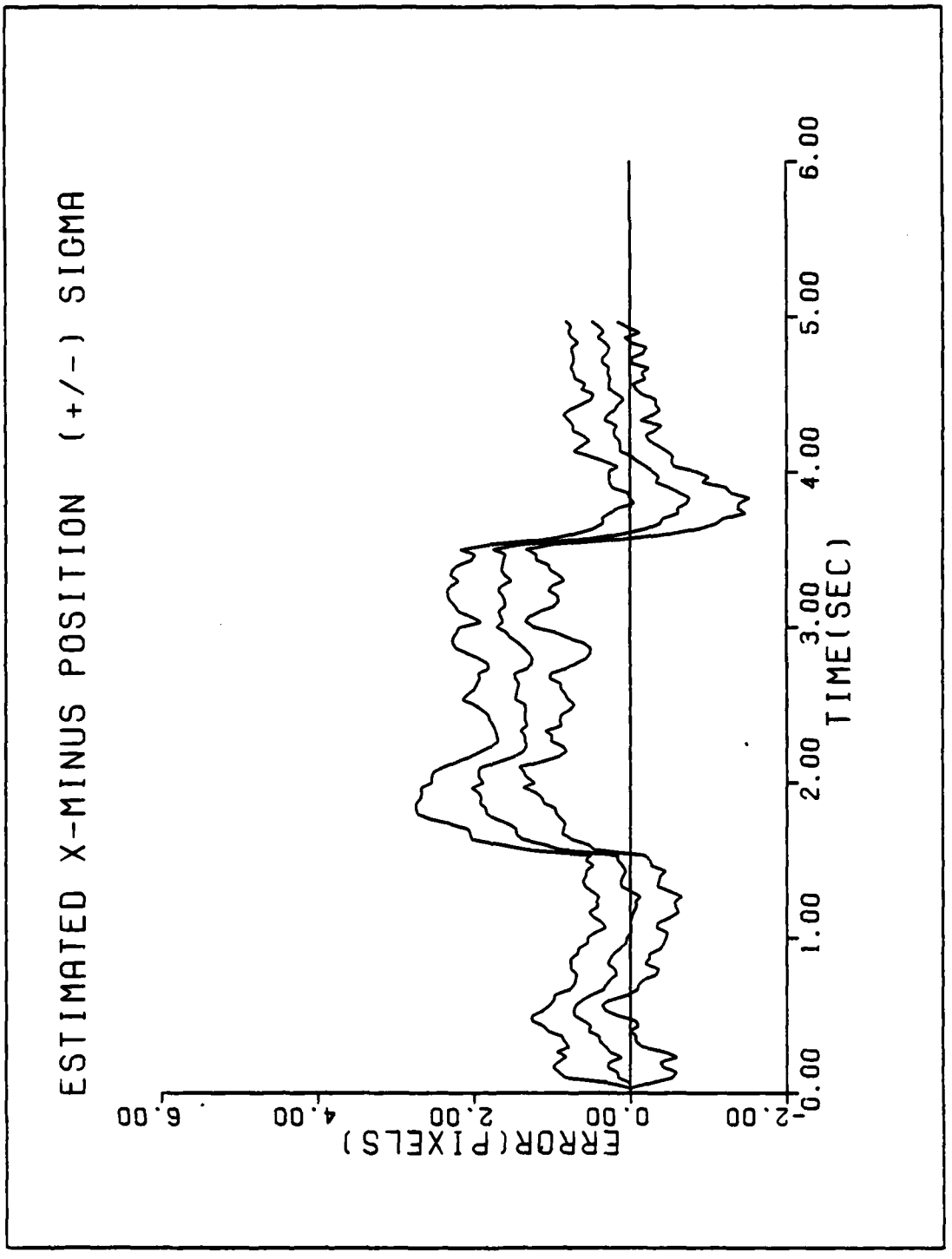


Figure B-8c. Performance Plot for T3G2MM-BAY

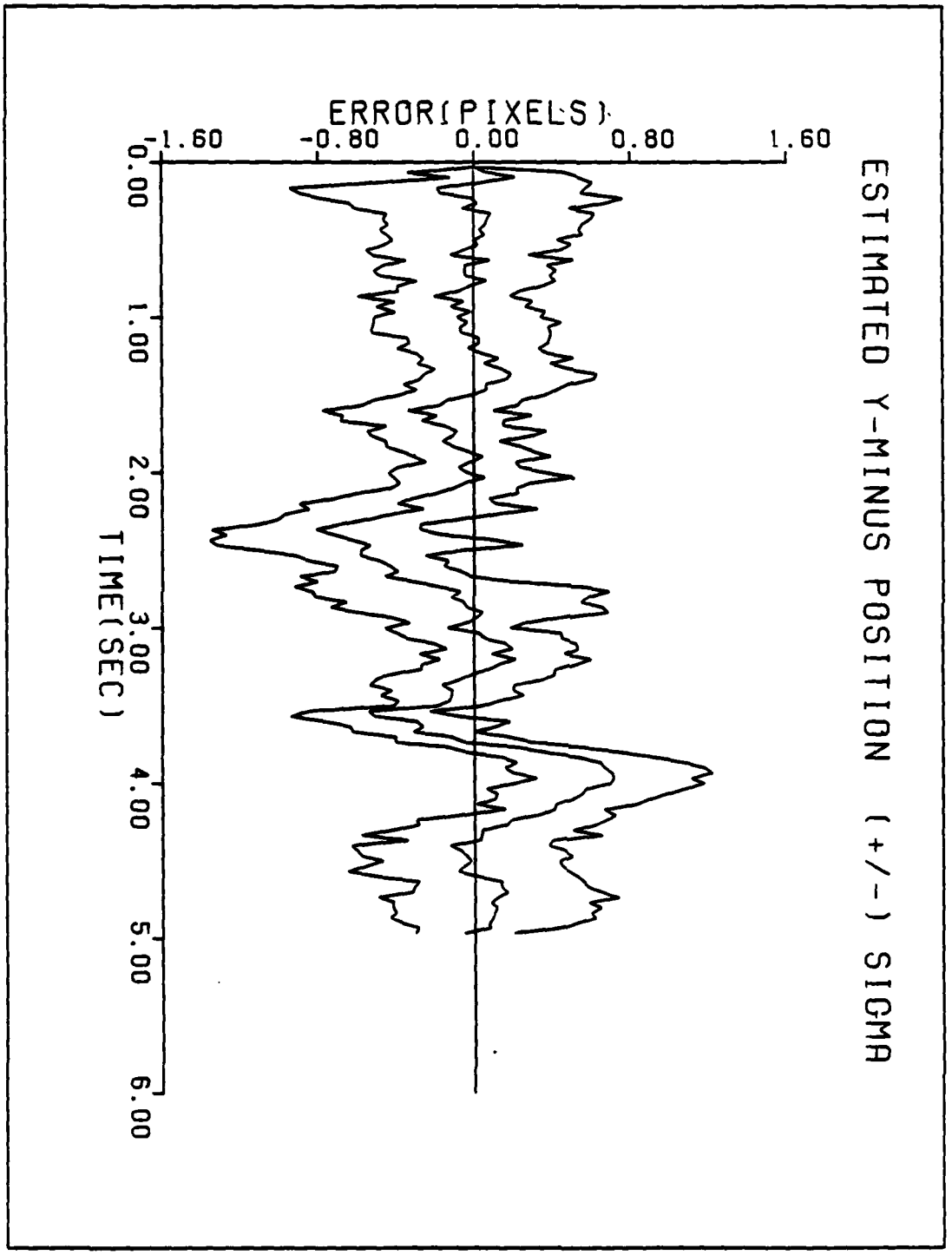


Figure B-8d. Performance Plot for TOGGIM-BAY

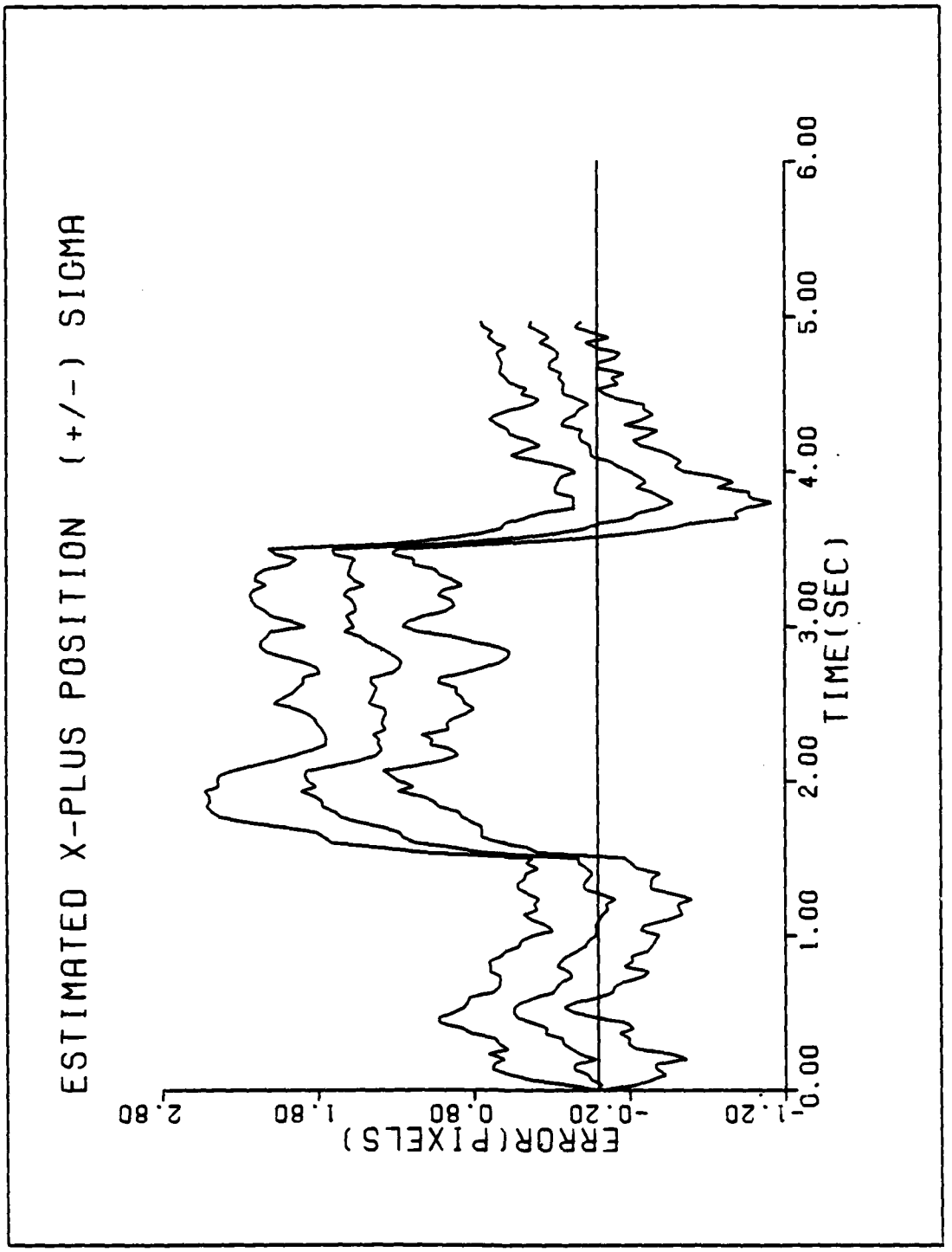


Figure B-8e. Performance Plot for T3G2MM-BAY

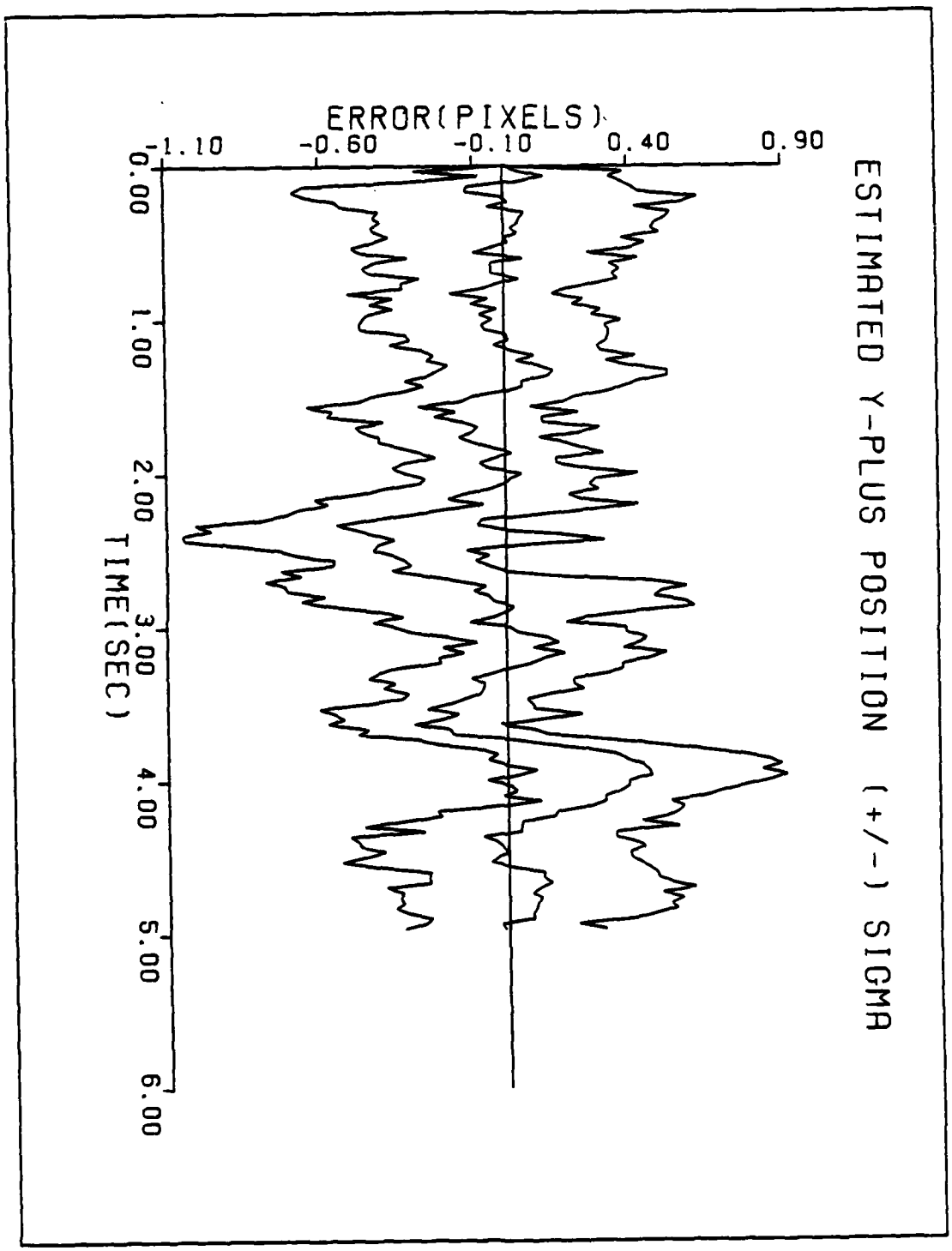


Figure B-8f. Performance Plot for TOGGMM-BAY

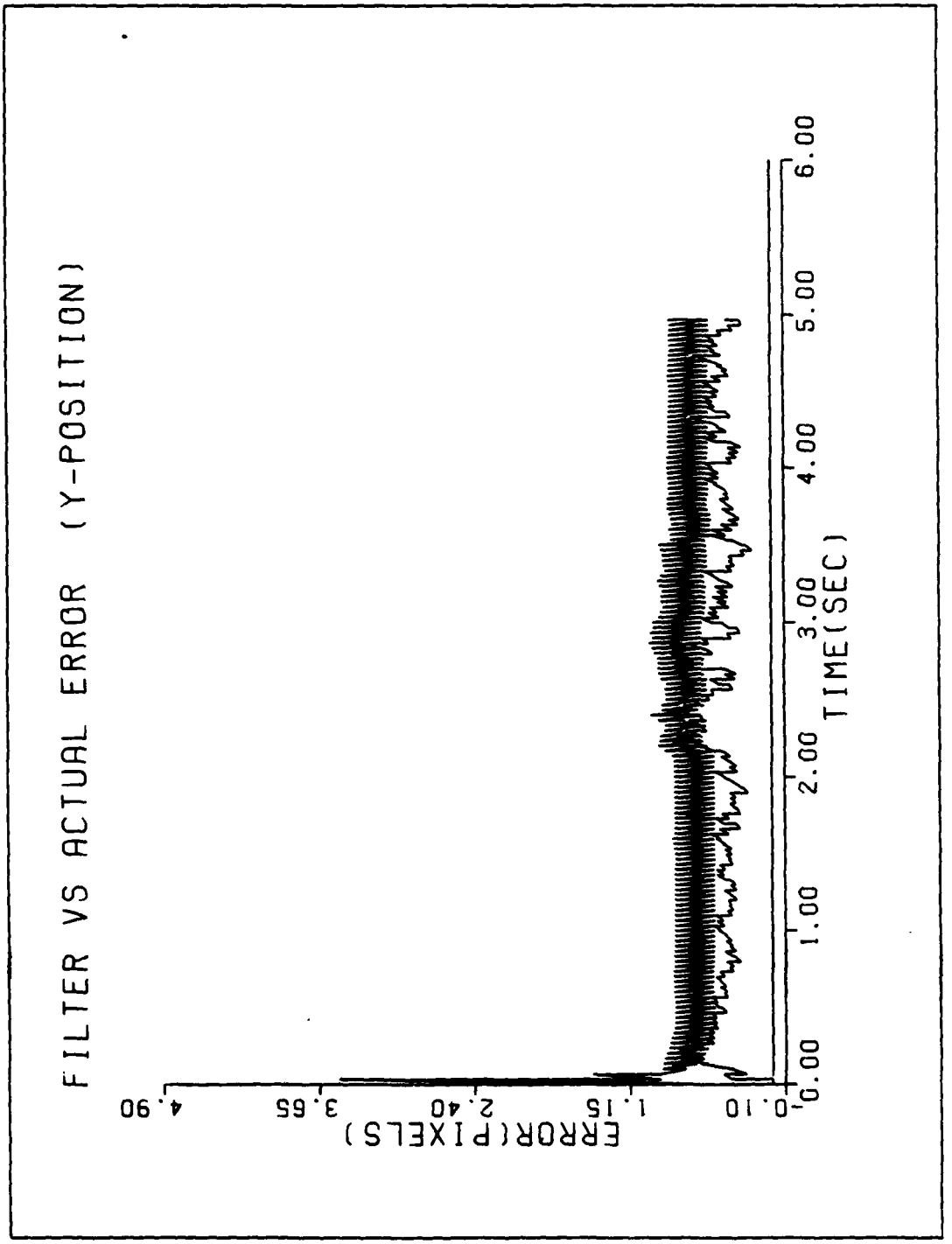


Figure B-9a. Performance Plot for T3G10MM-BAY

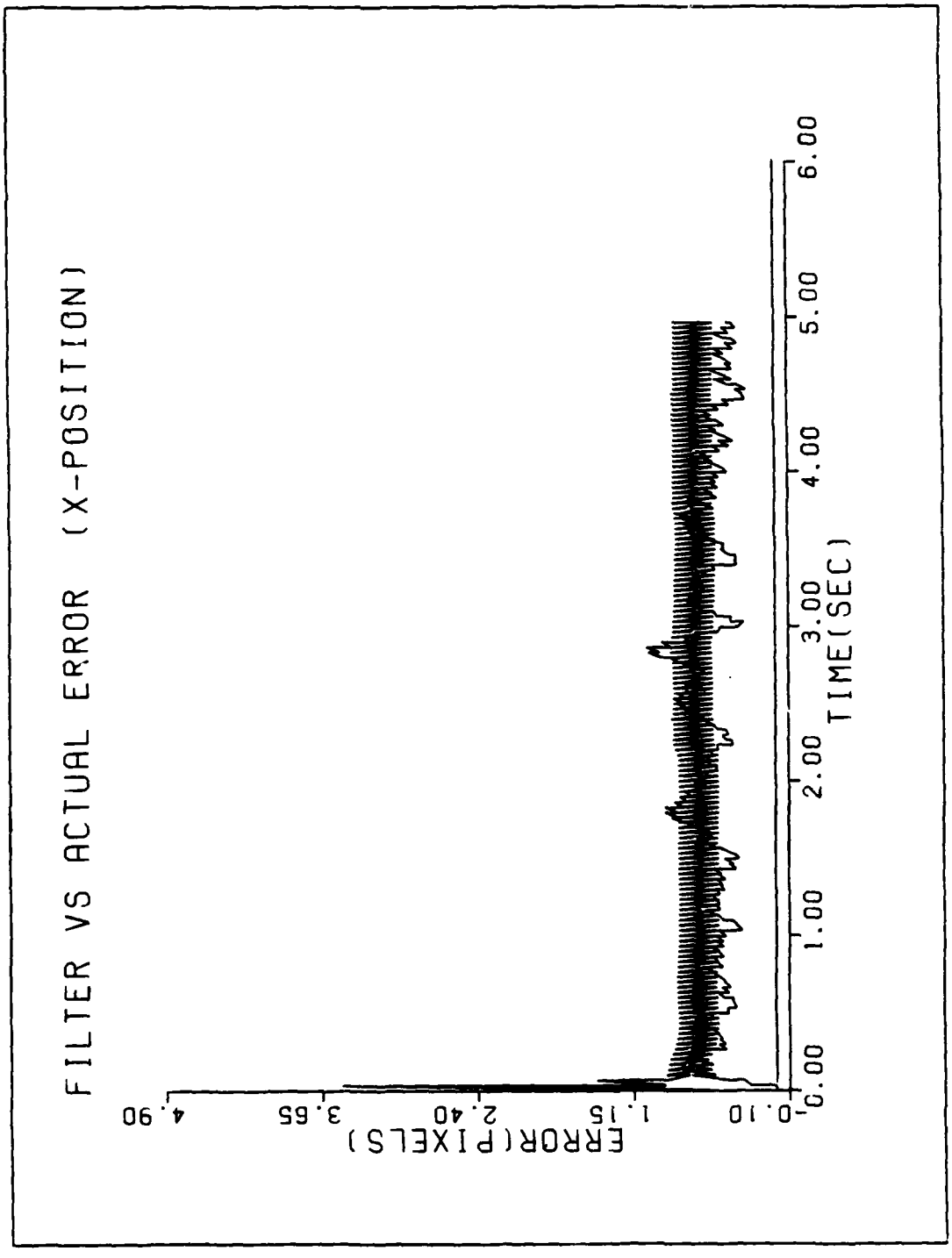


Figure B-9b. Performance Plot for T3G10MM-BAY

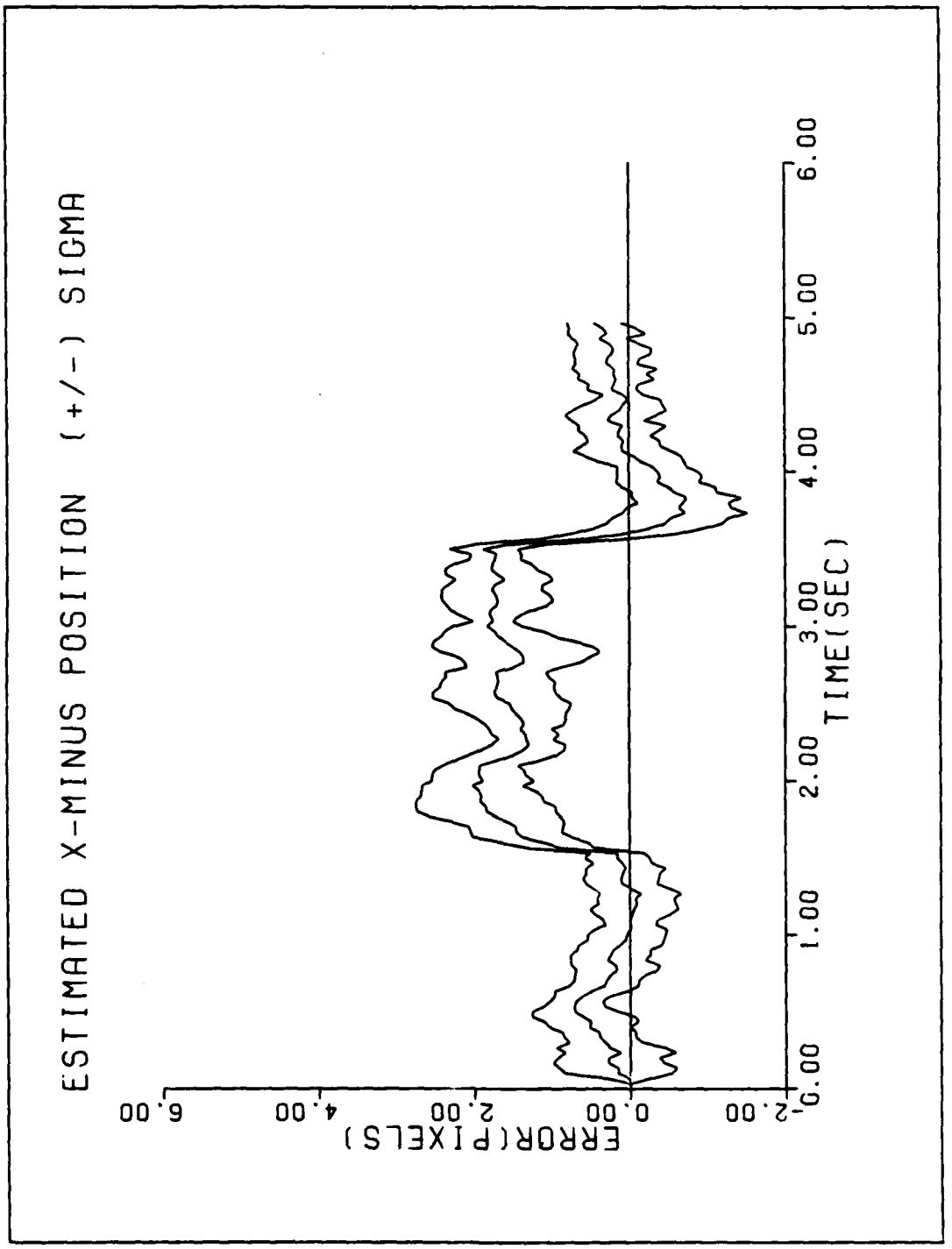


Figure B-9c. Performance Plot for T3G10MM-BAY

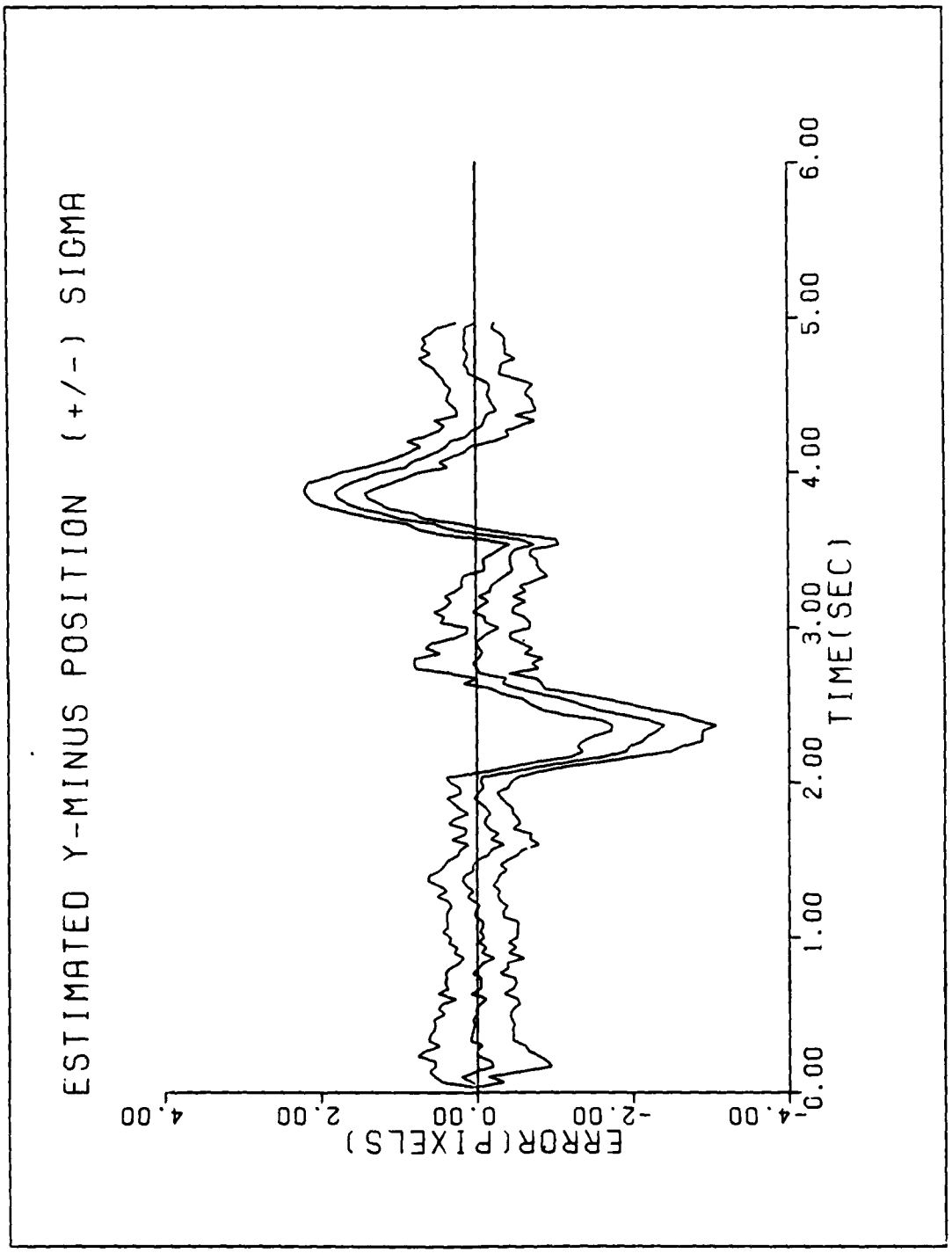


Figure B-9d. Performance Plot for T3G10MM-BAY

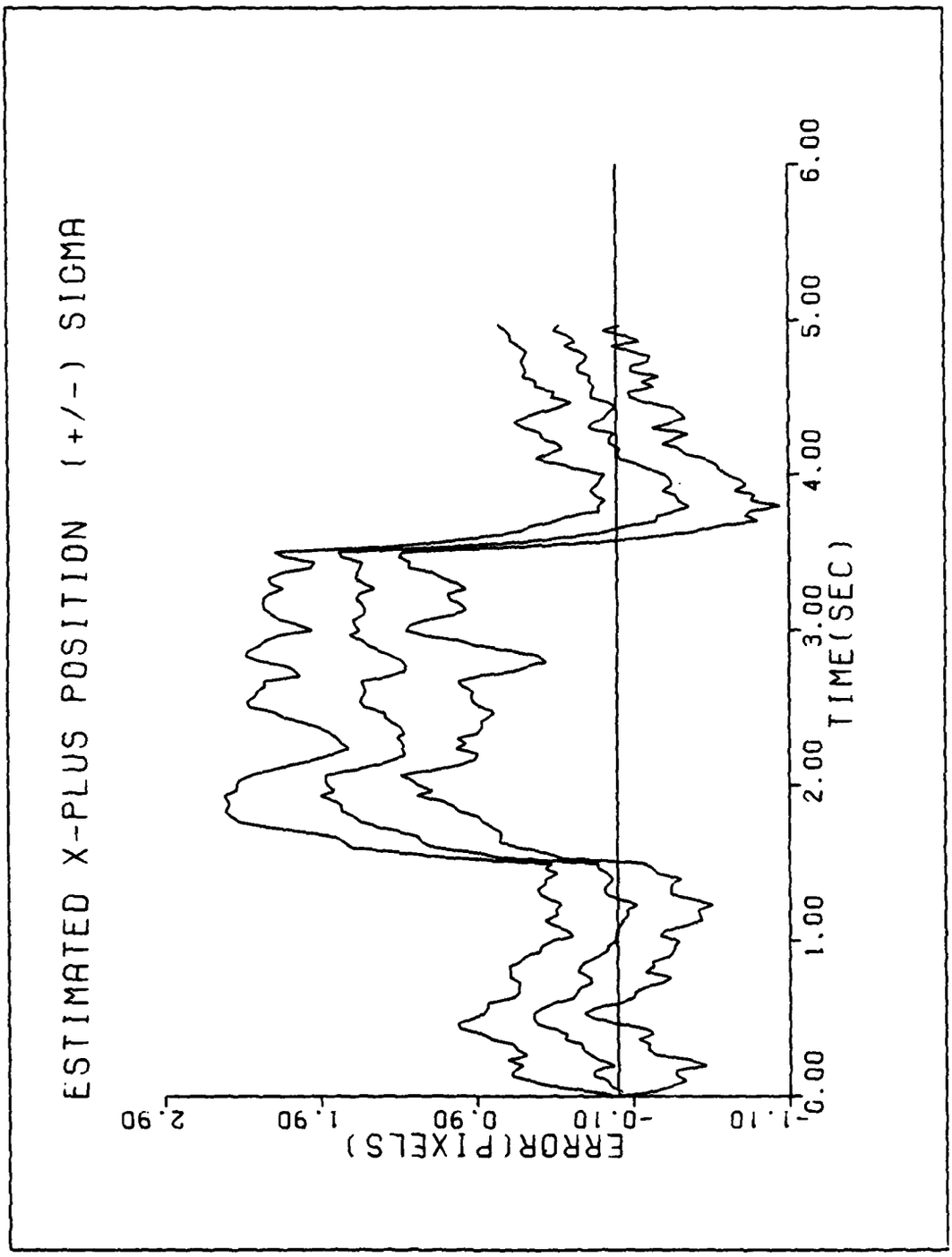


Figure B-9e. Performance Plot for T3G10MM-BAY

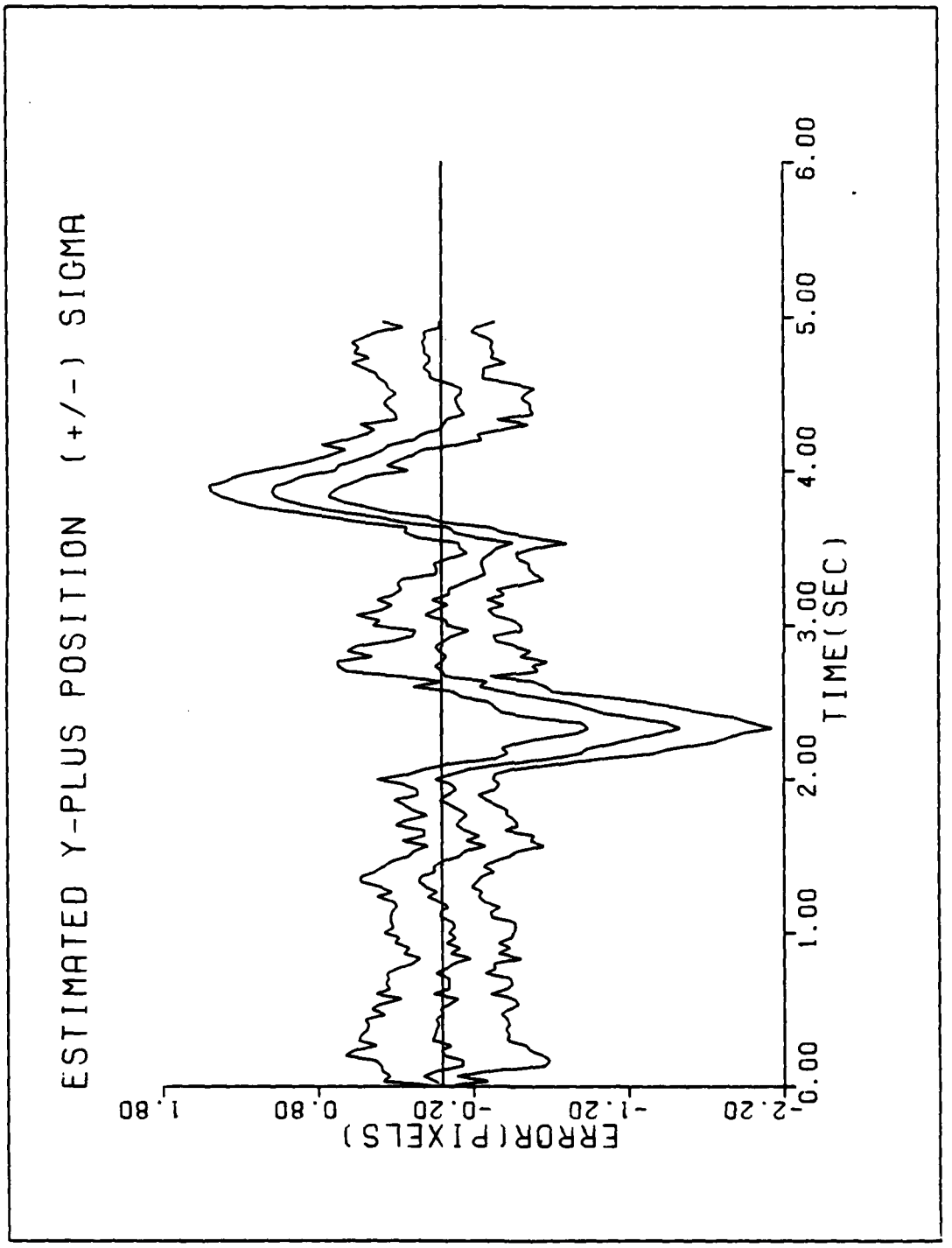


Figure B-9f. Performance Plot for T3G10MM-BAY

APPENDIX C

Performance Plots for the MAP MMAF

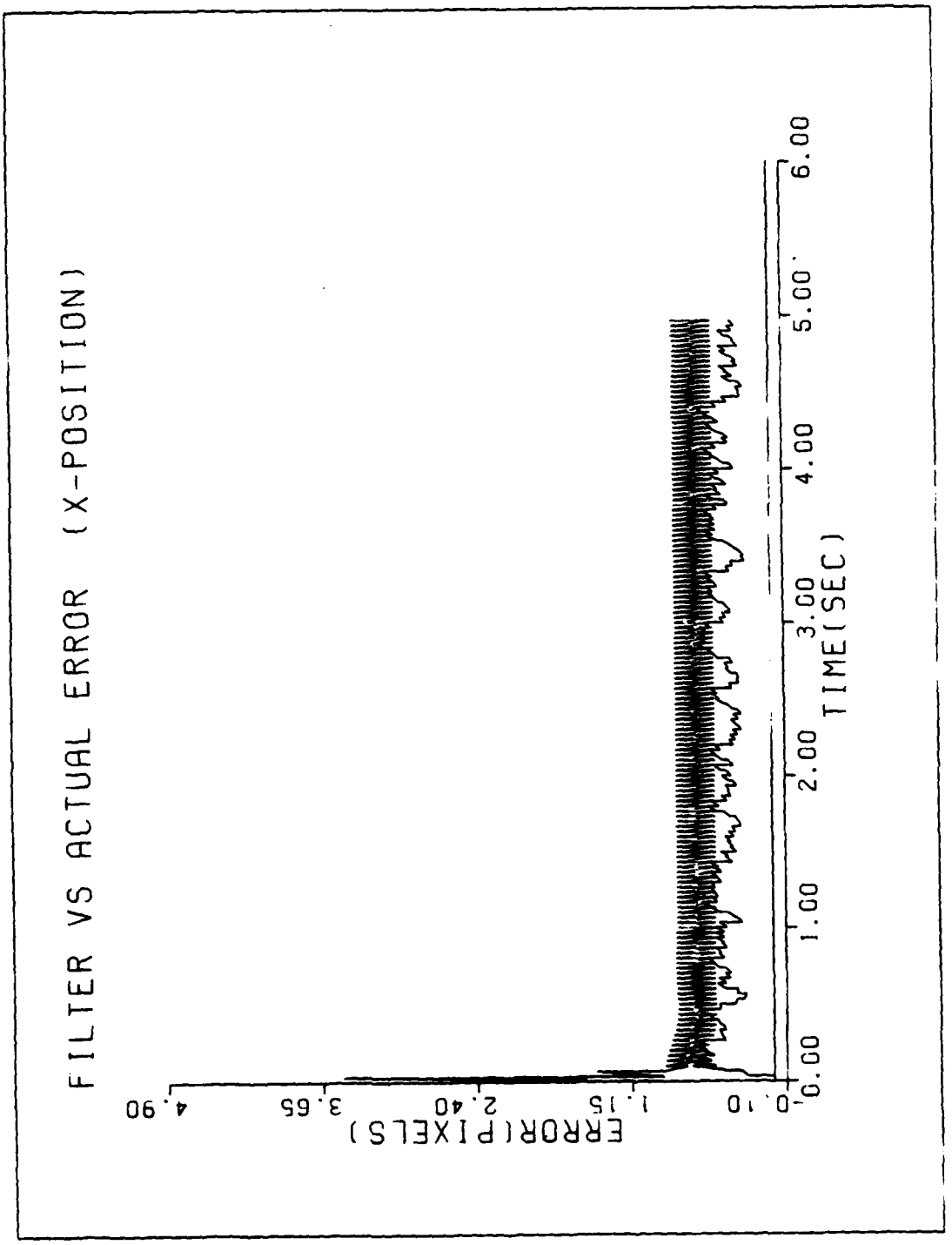


Figure C-1a. Performance Plot for TIMM-MAP

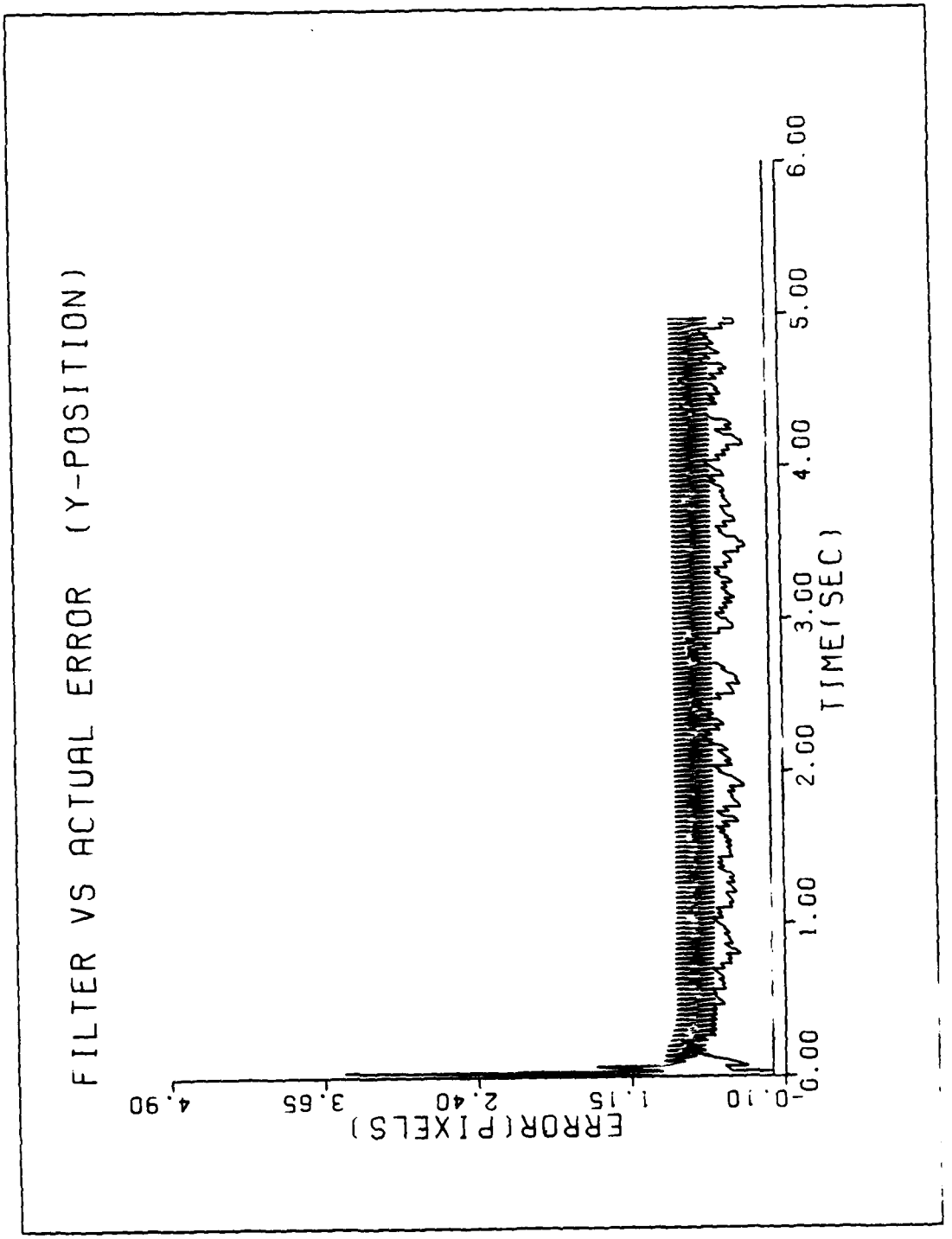


Figure C-1b. Performance Plot for TIMM-MAP

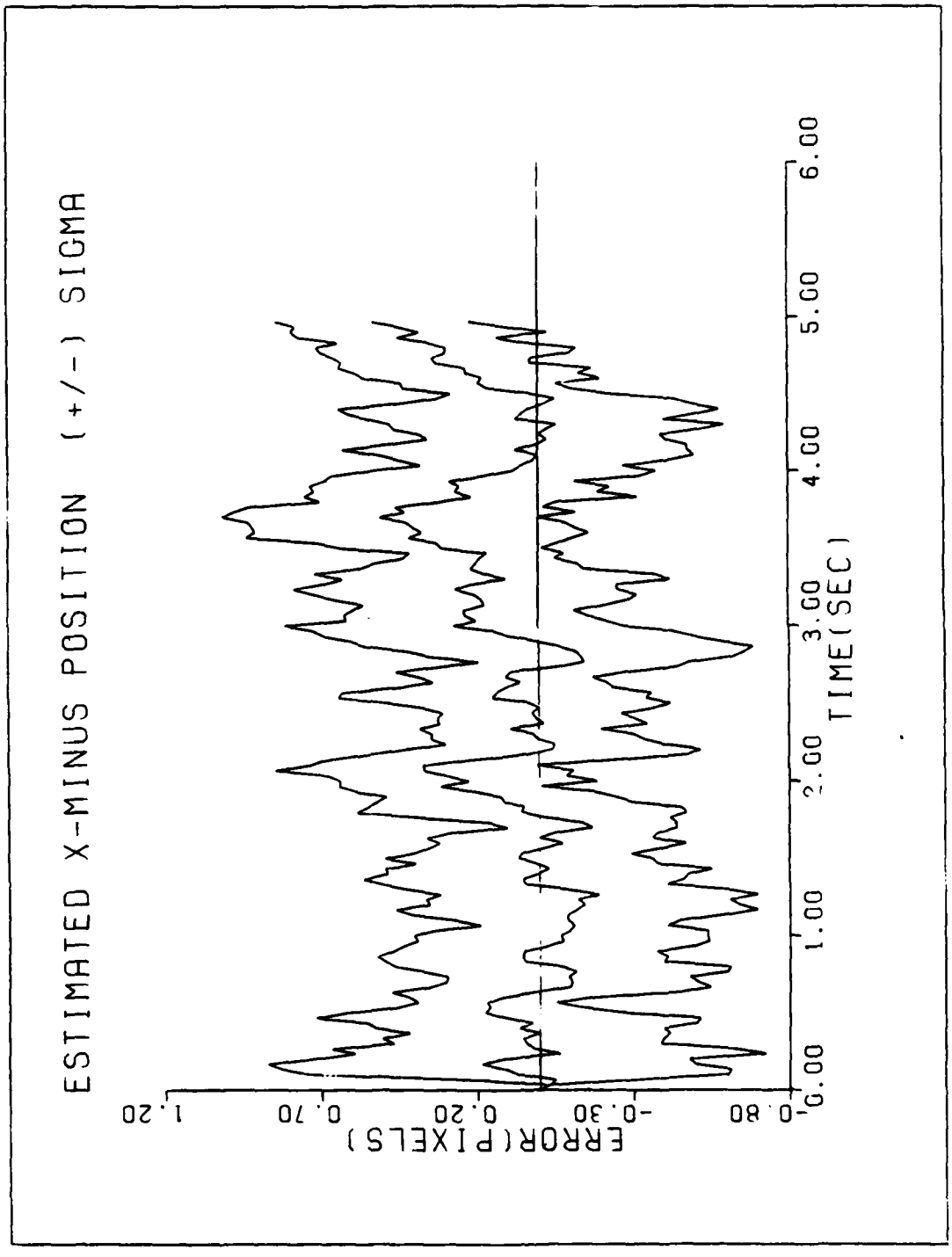


Figure C-1c. Performance Plot for TIMM-MAP

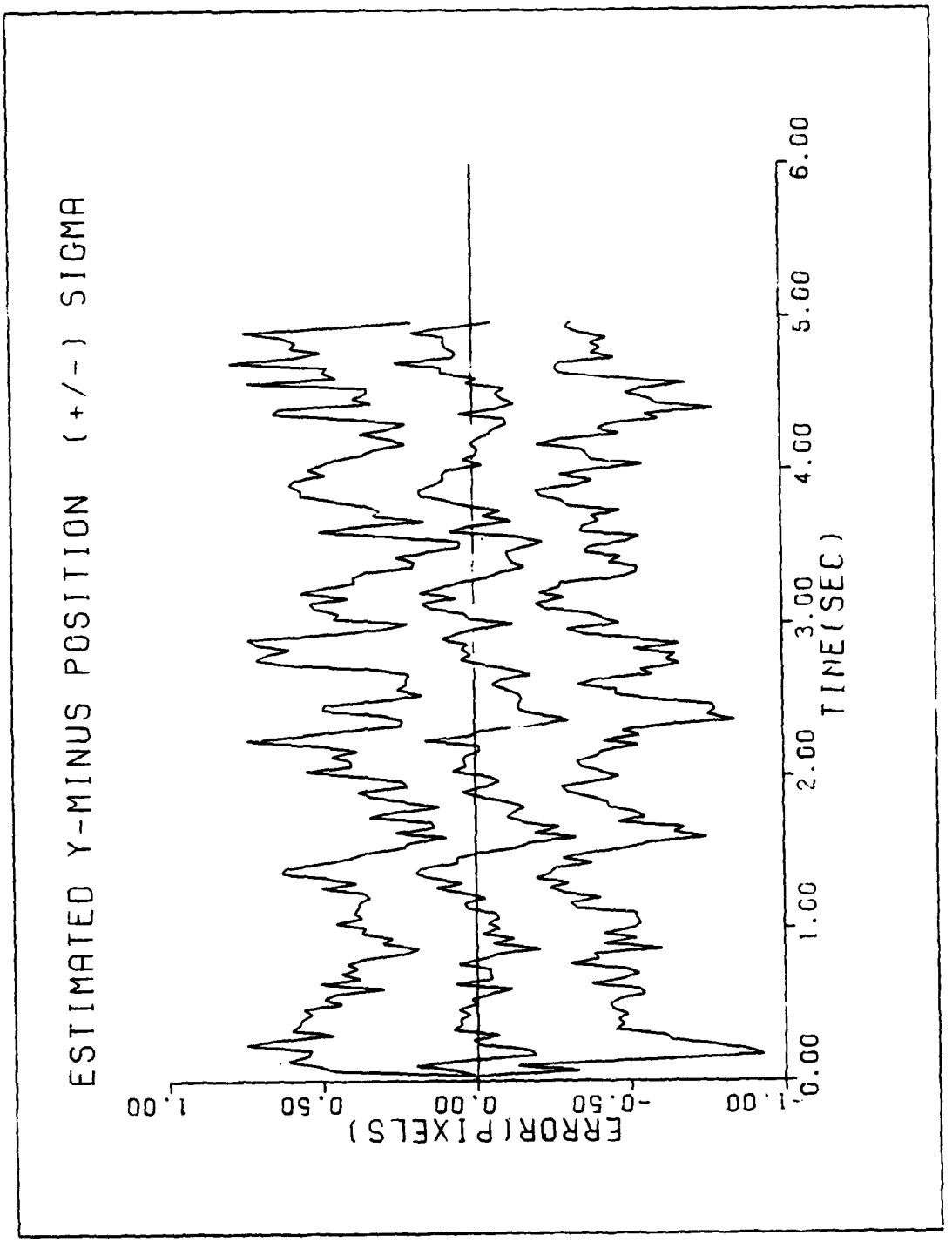


Figure C-1d. Performance Plot for TIMM-MAP

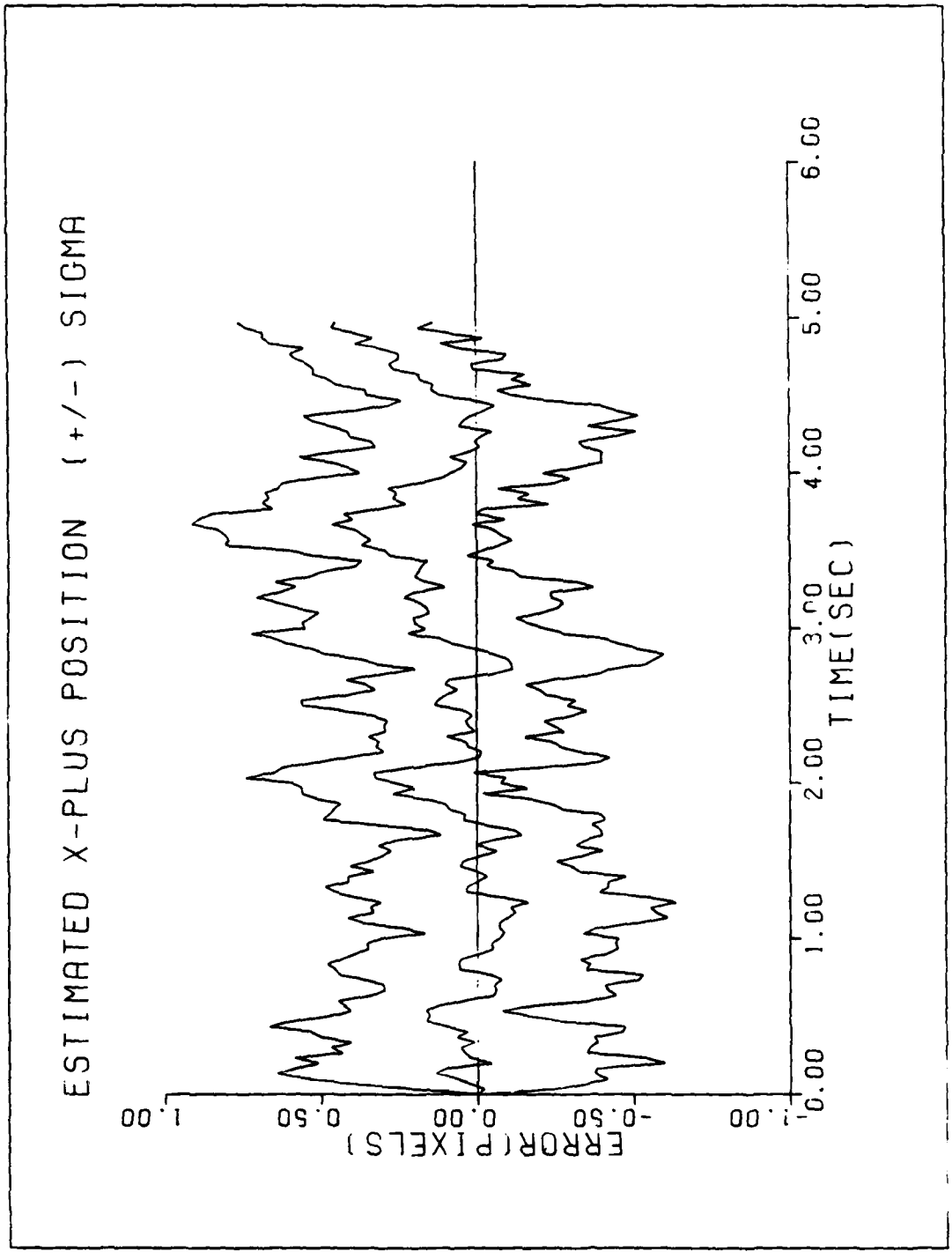


Figure C-1e. Performance Plot for TIMM-MAP

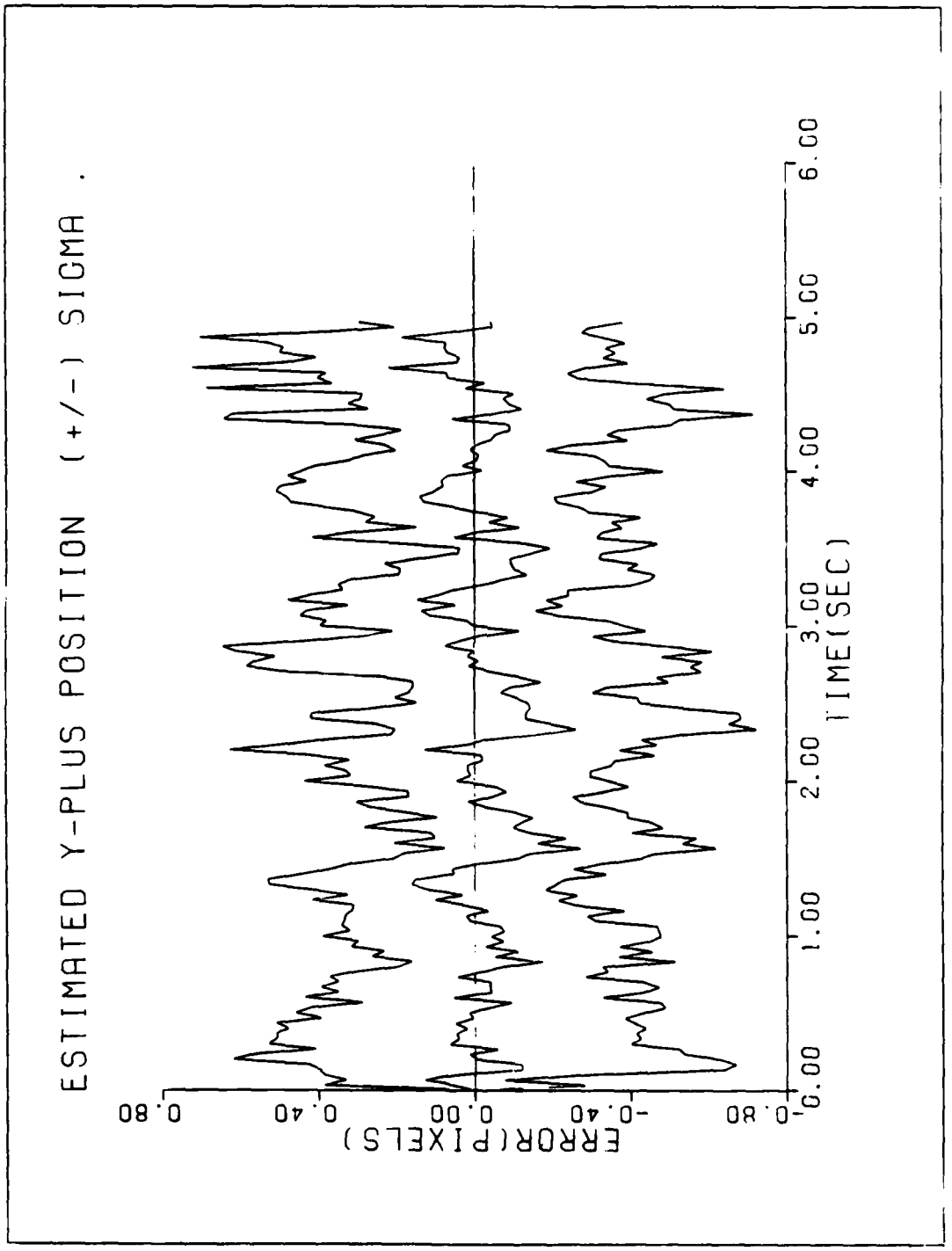


Figure C-1f. Performance Plot for TIMM-MAP

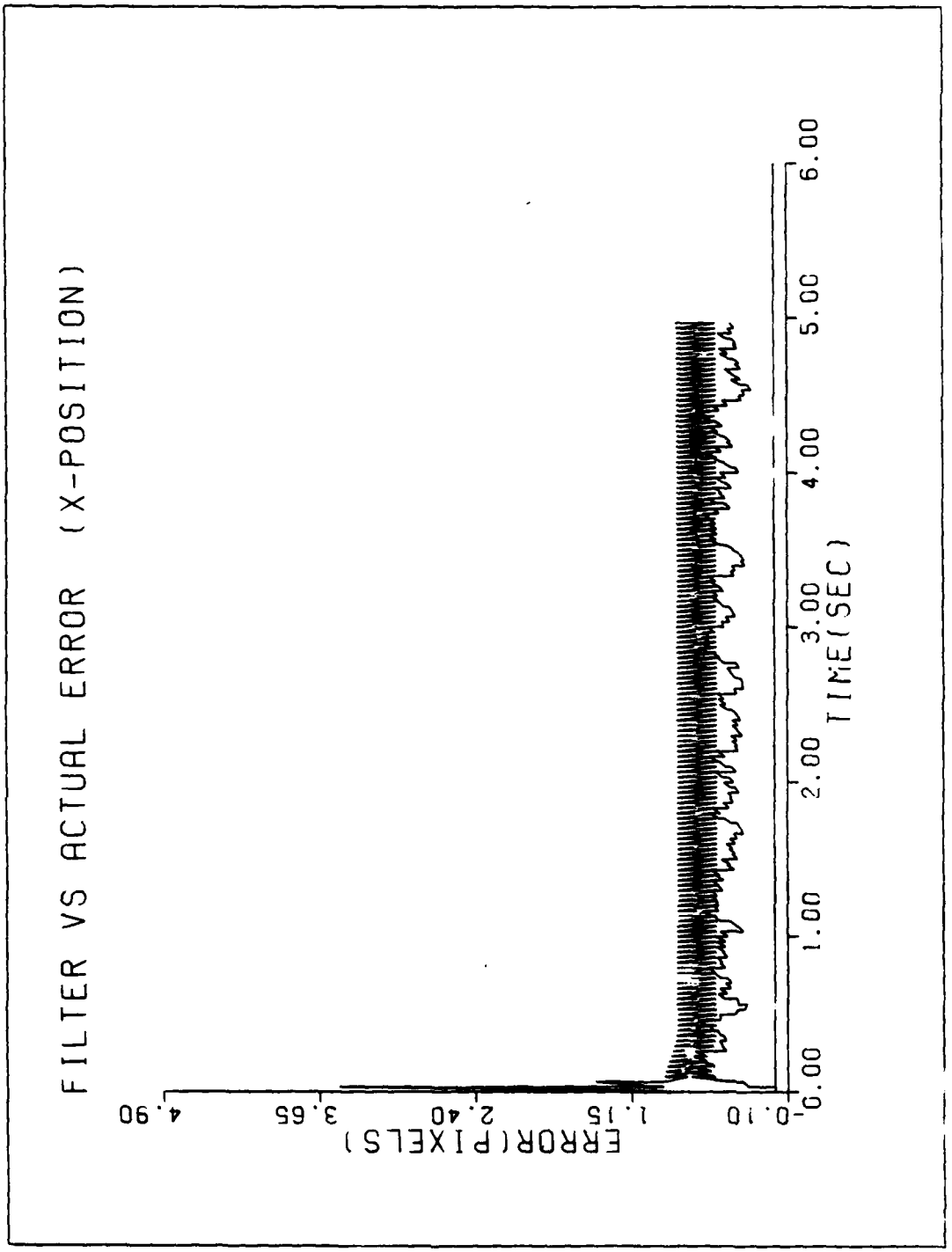


Figure C-2a. Performance Plot for T2G2MM-MAP

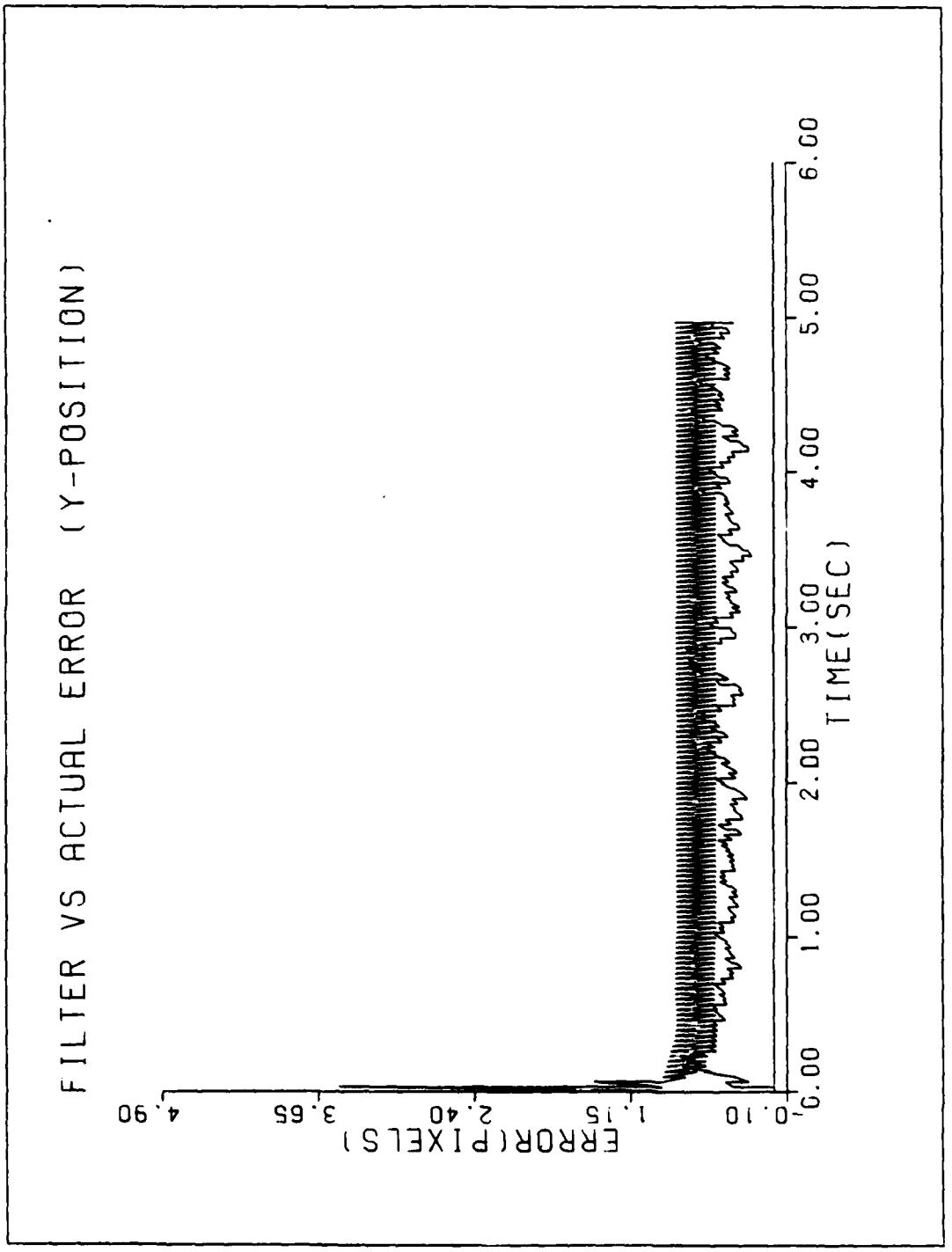


Figure C-2b. Performance Plot for T2G2MM-MAP

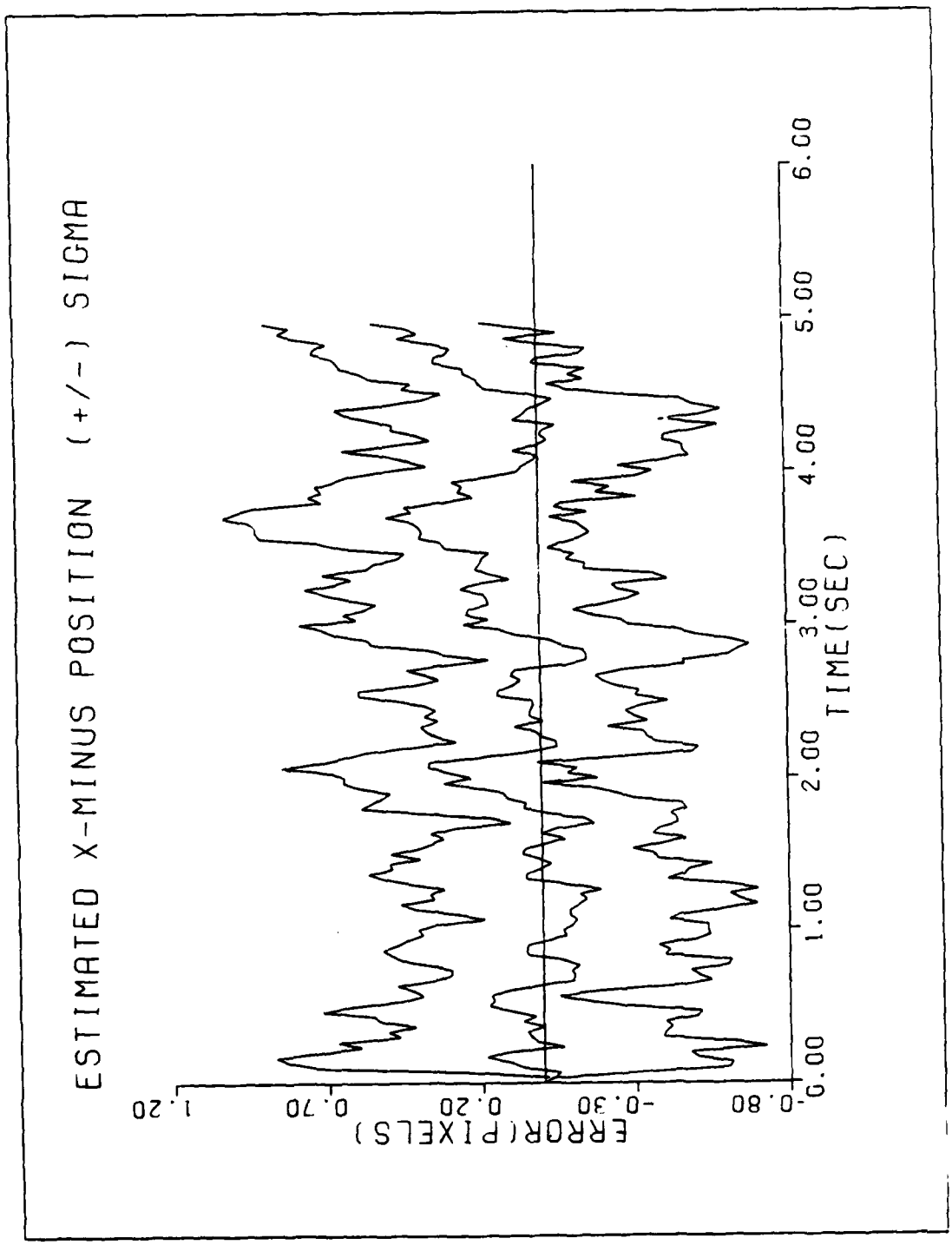


Figure C-2c. Performance Plot for T2G2MM-MAP

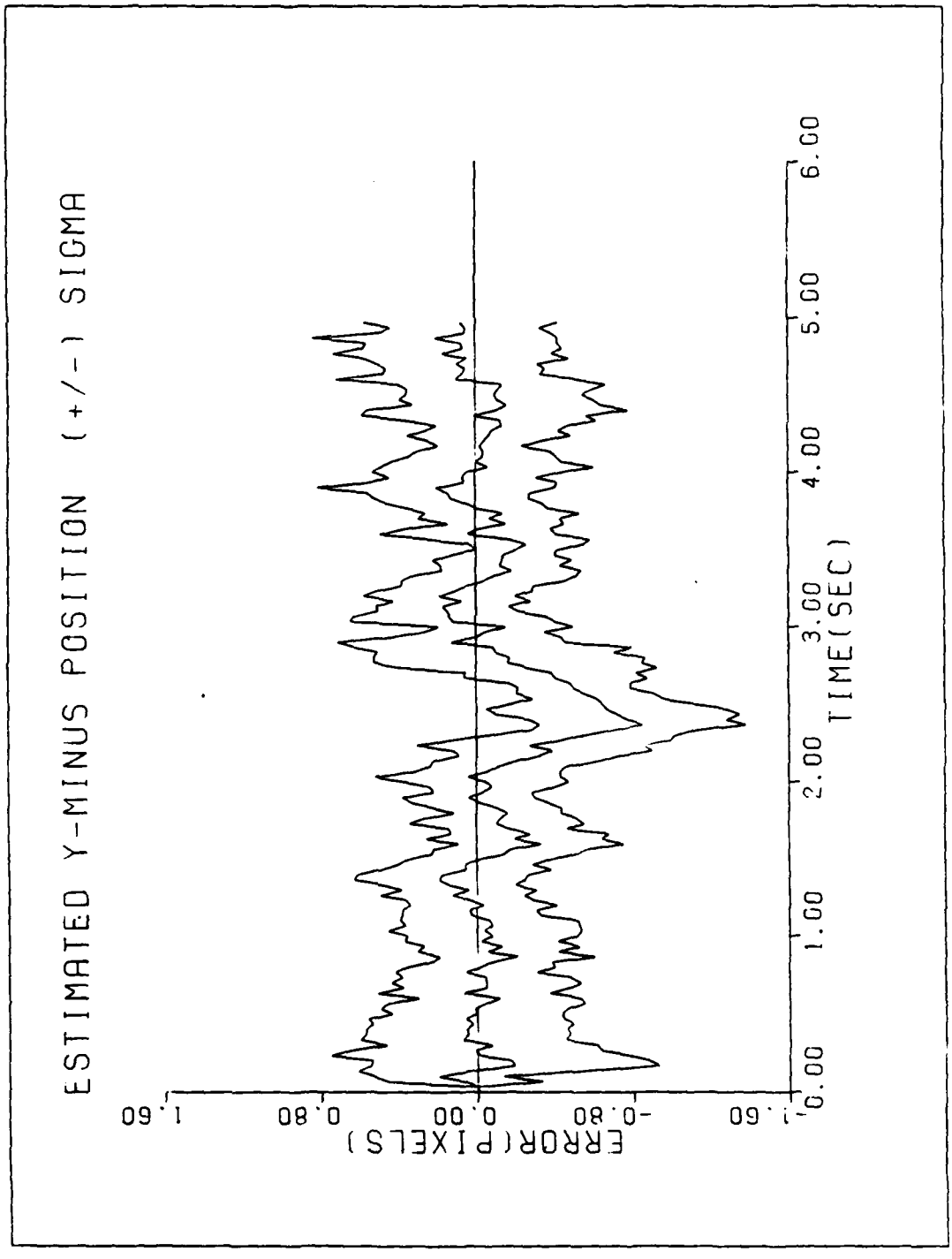


Figure C-2d. Performance Plot for T2G2MM-MAP

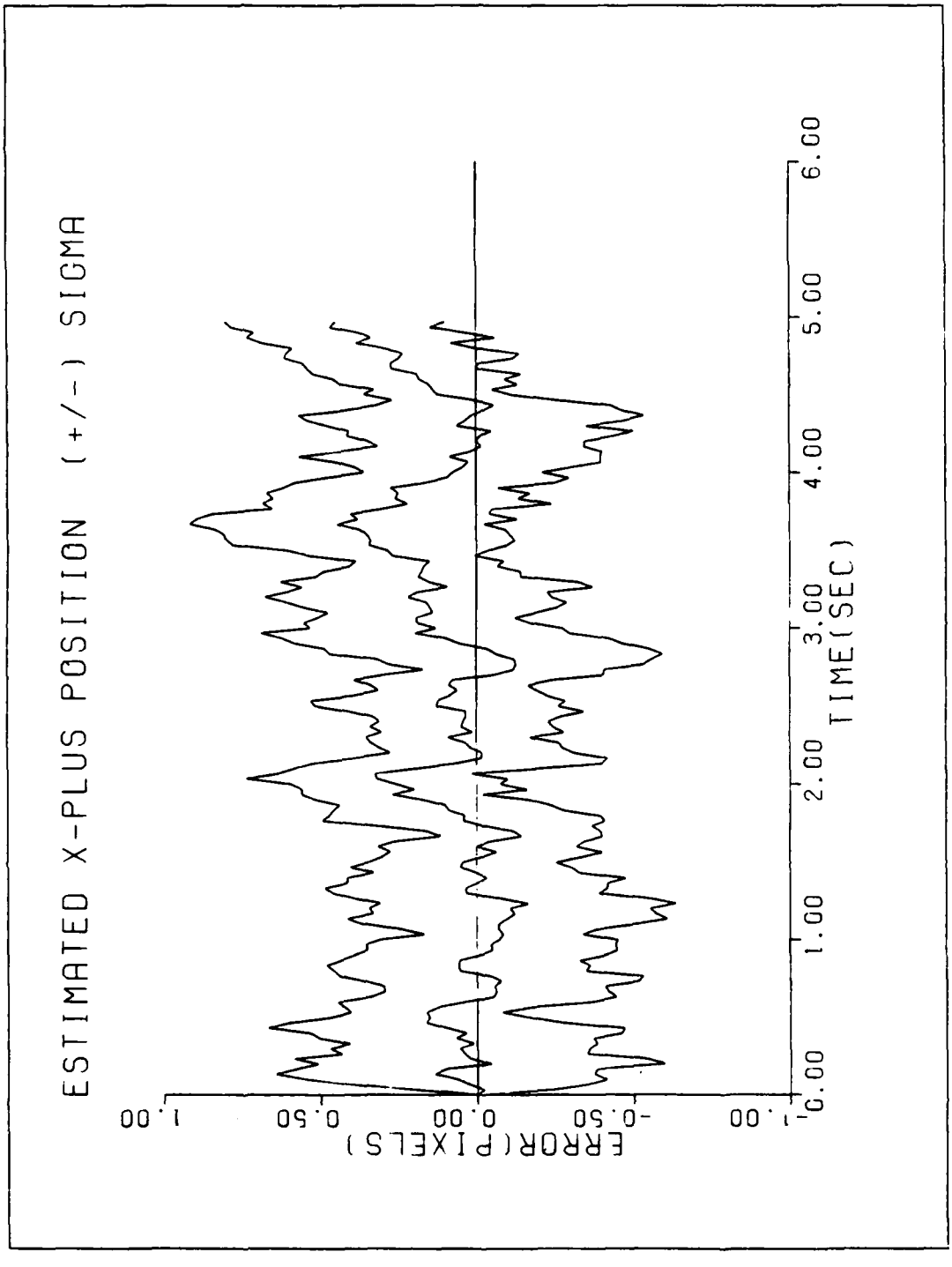


Figure C-2e. Performance Plot for T2G2MM-MAP

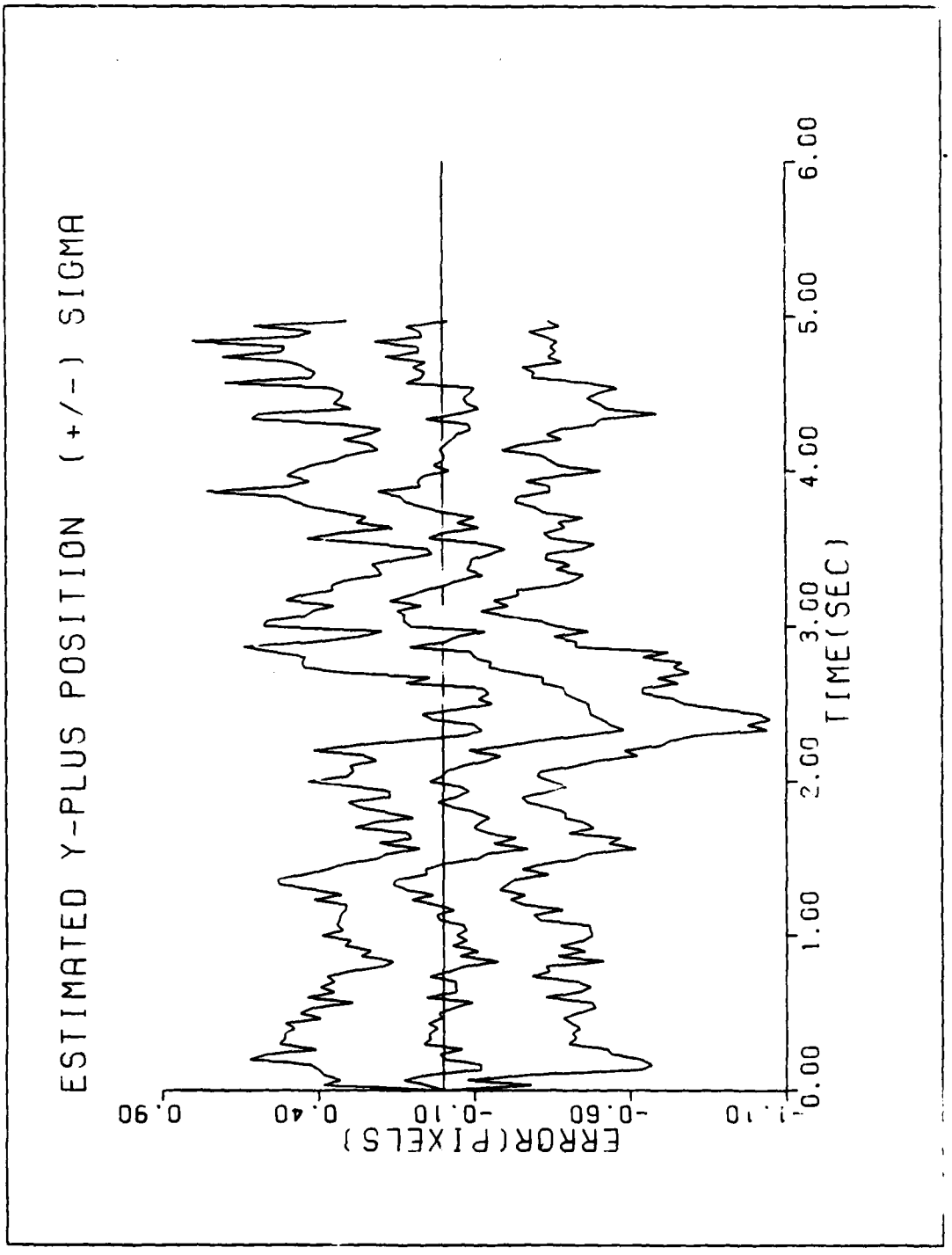


Figure C-2f. Performance Plot for T2G2MM-MAP

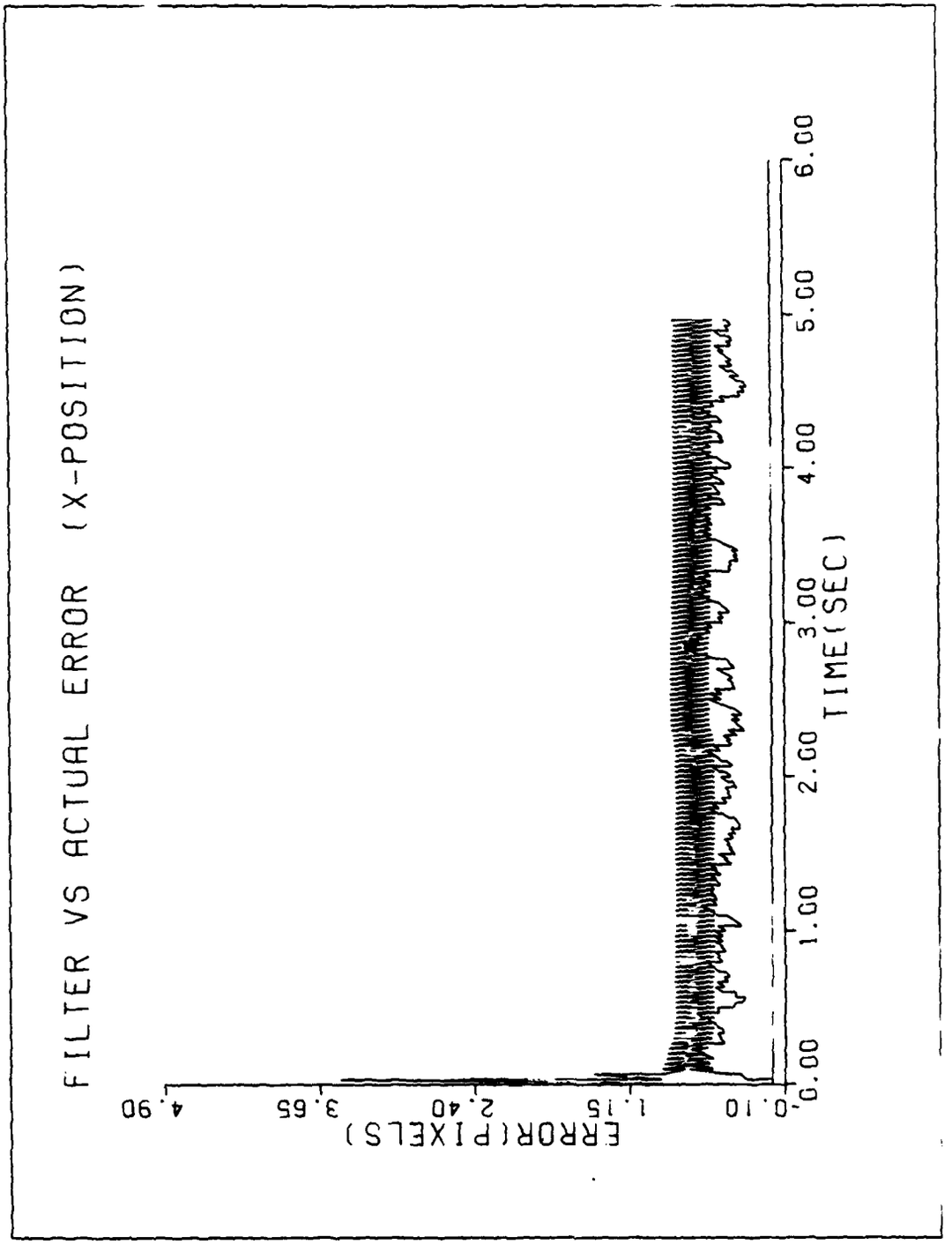


Figure C-3a. Performance Plot for T2G10MM-MAP

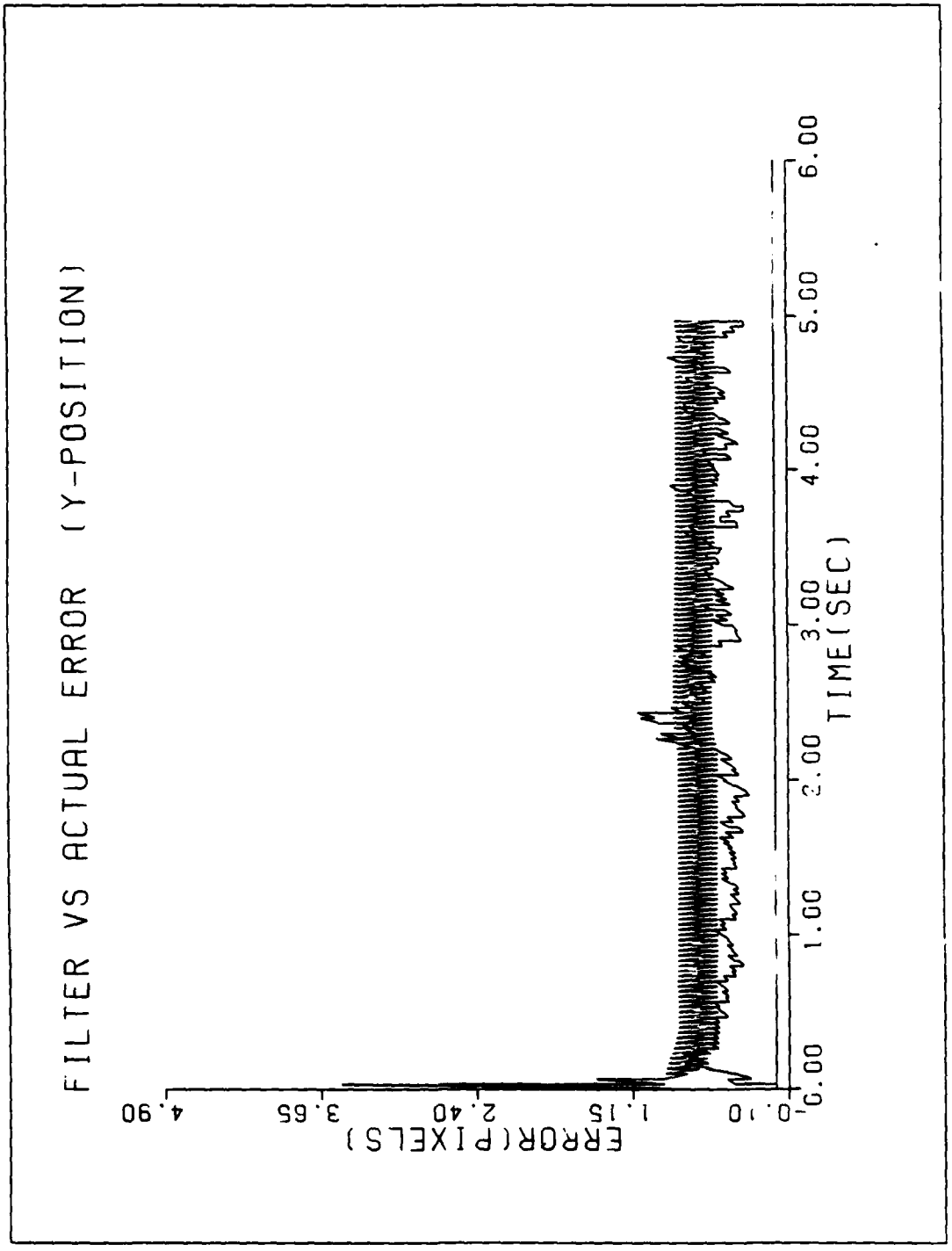


Figure C-3b. Performance Plot for T2G10MM-MAP

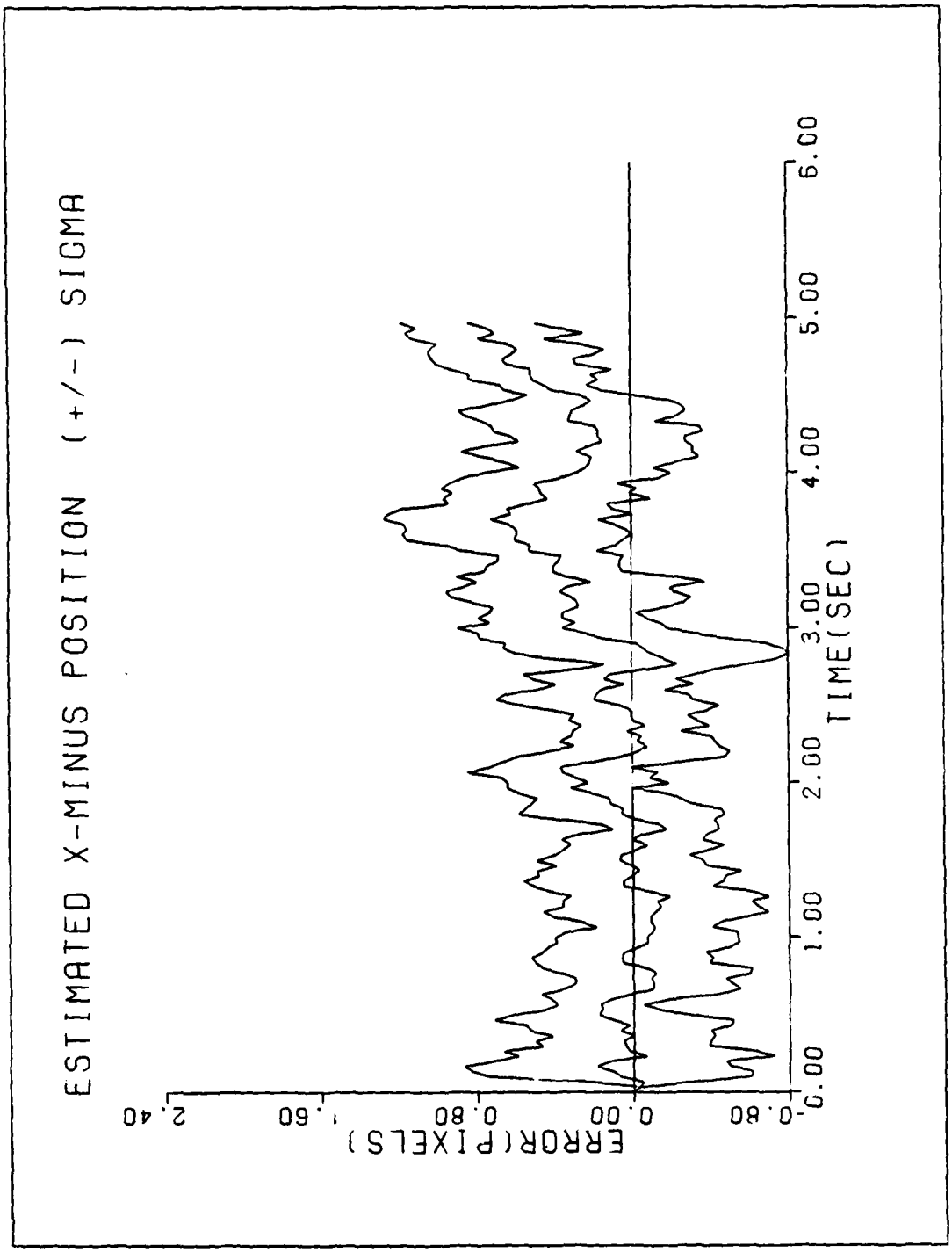


Figure C-3c. Performance Plot for T2G10MM-MAP

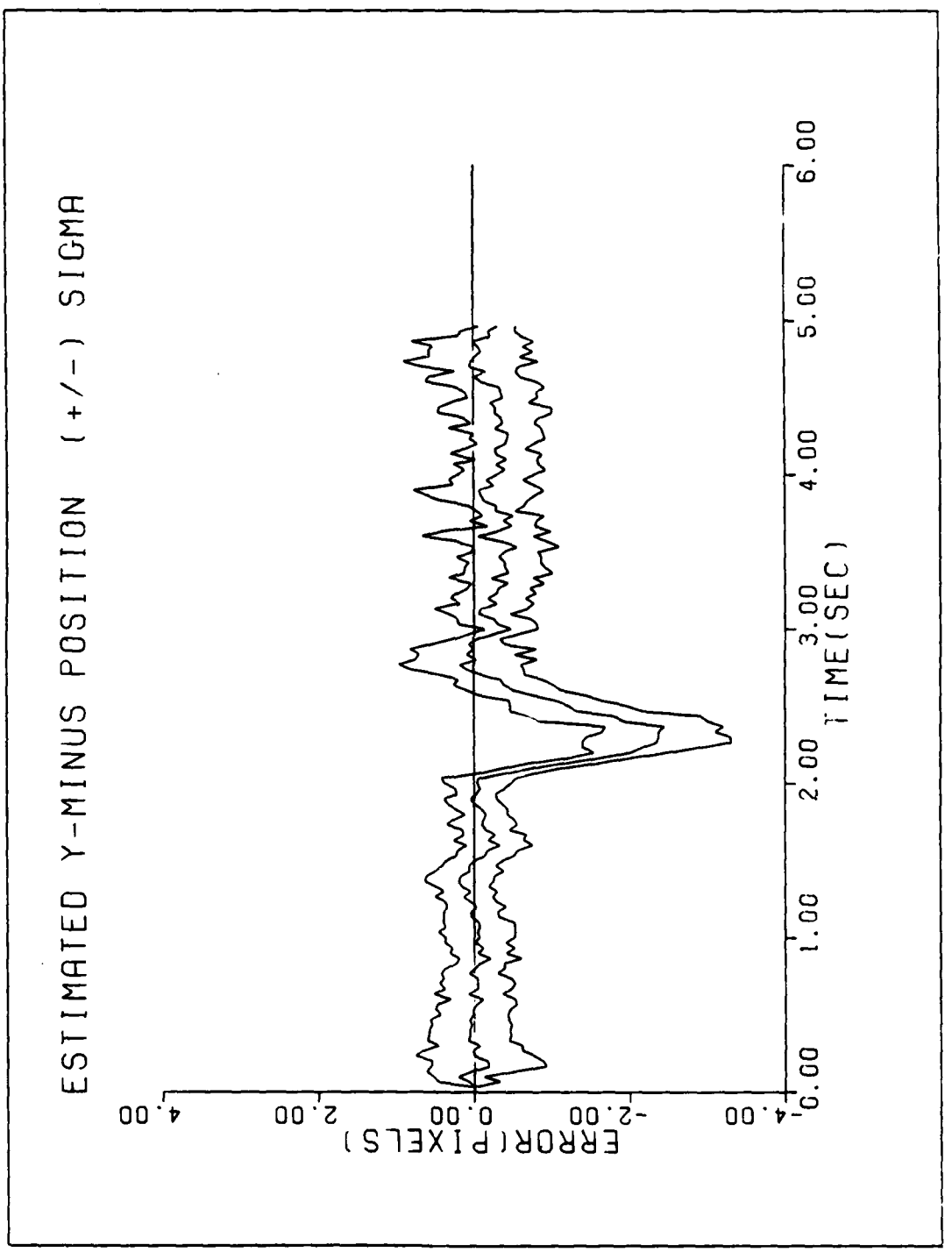


Figure C-3d. Performance Plot for T2G10MM-MAP

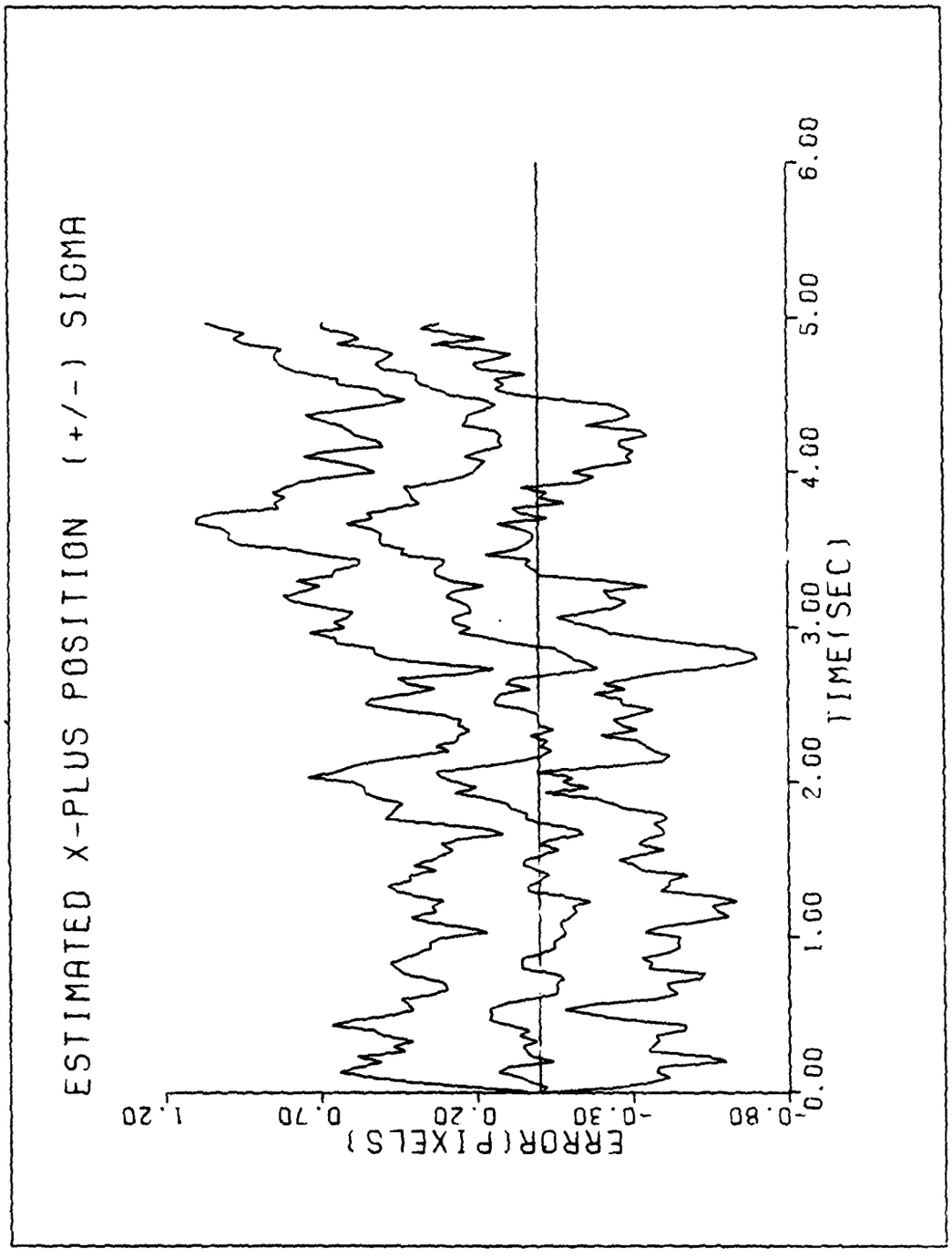


Figure C-3e. Performance Plot for T2G10MM-MAP

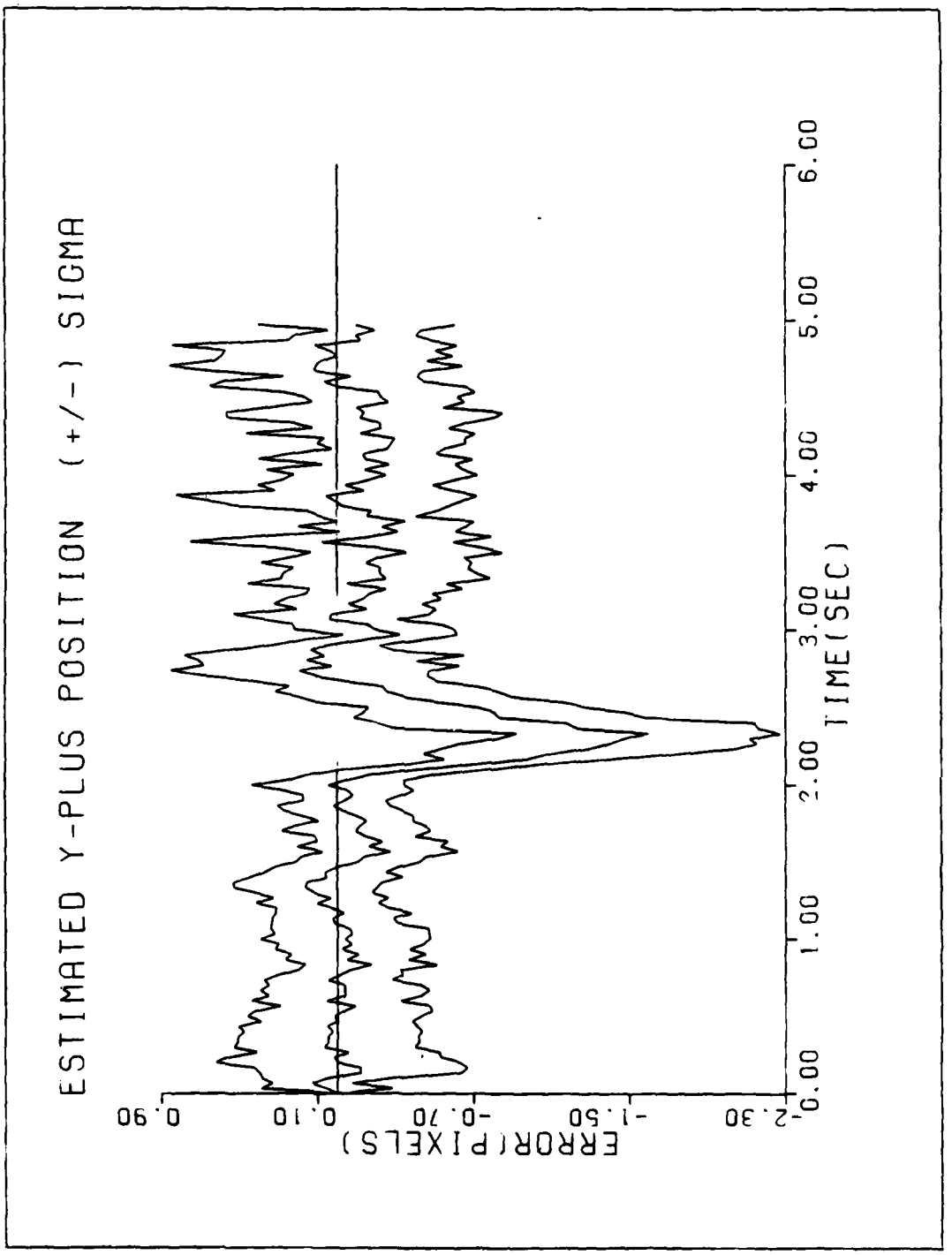


Figure C-3f. Performance Plot for T2G10MM-MAP

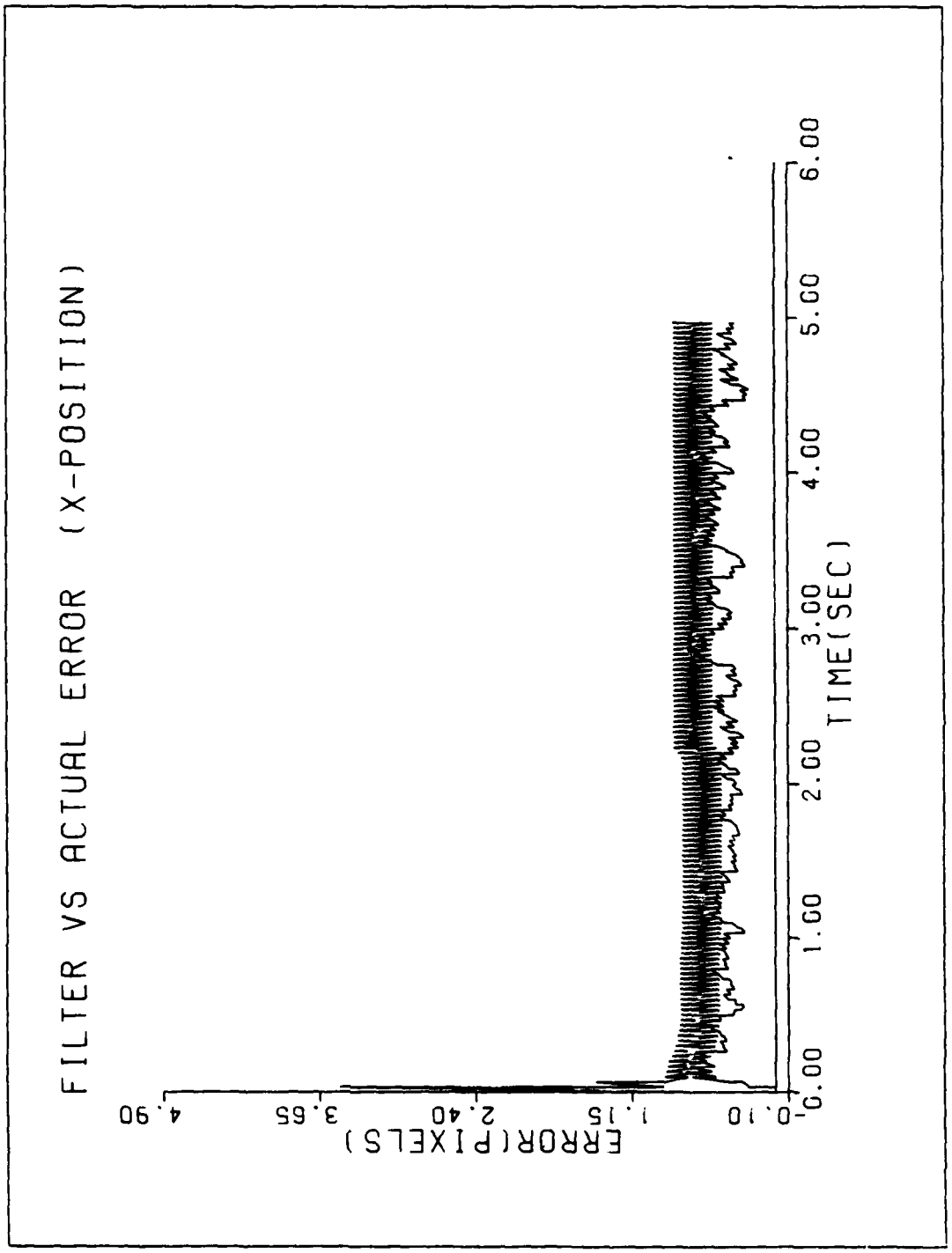


Figure C-4a. Performance Plot for T2G10MM-MAP (Retuned)

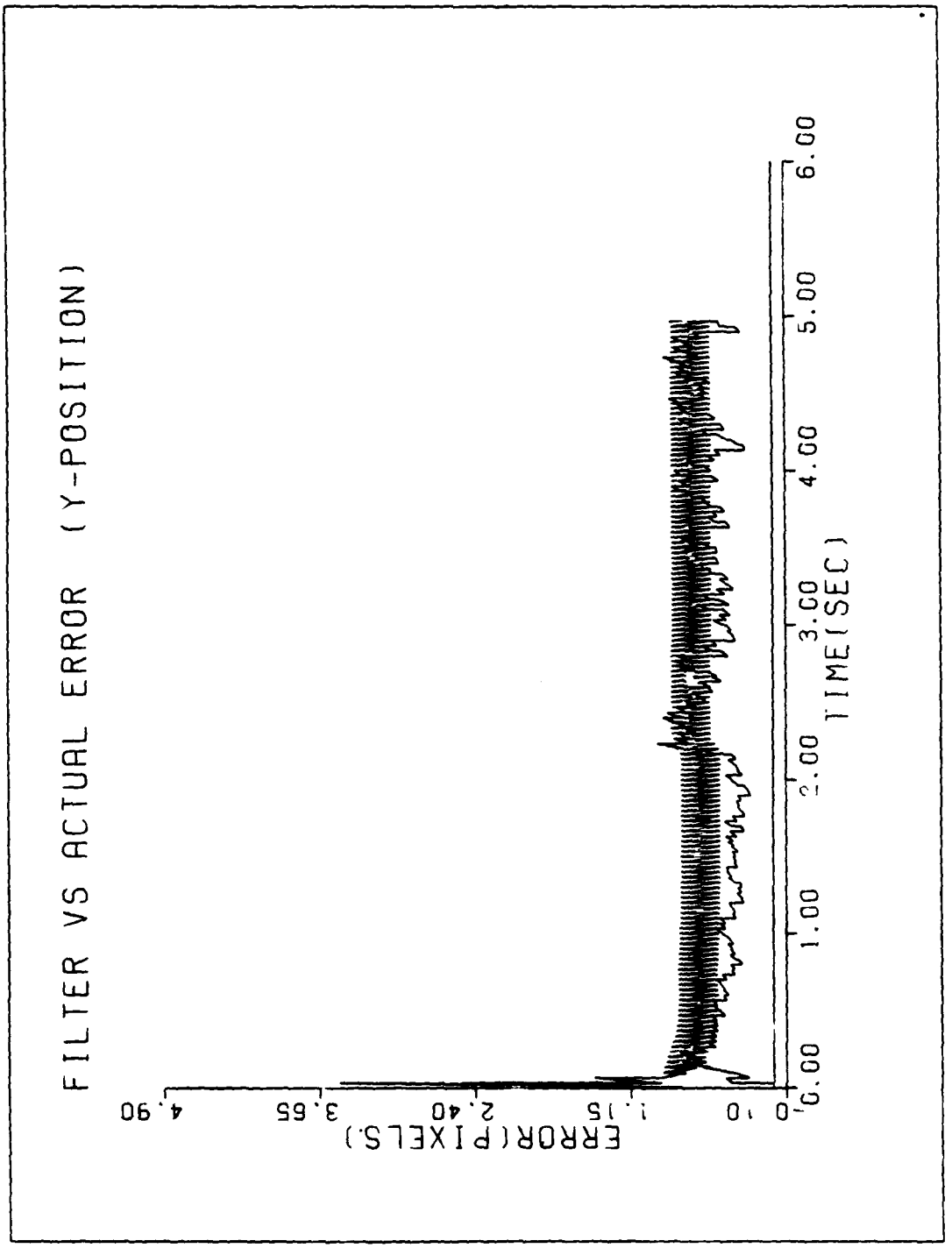


Figure C-4b. Performance Plot for T2G10MM-MAP (Retuned)

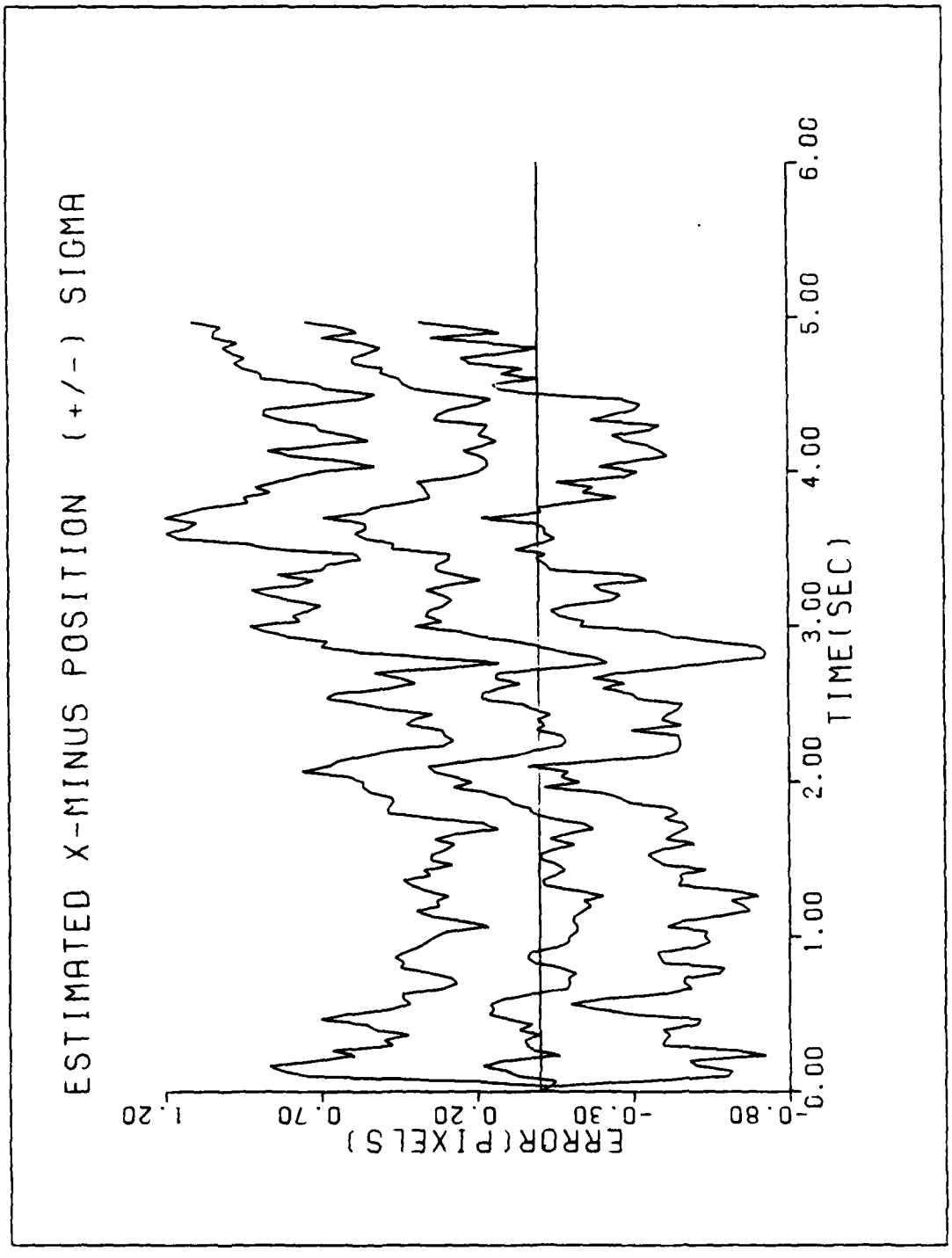


Figure C-4c. Performance Plot for T2G10MM-MAP (Retuned)

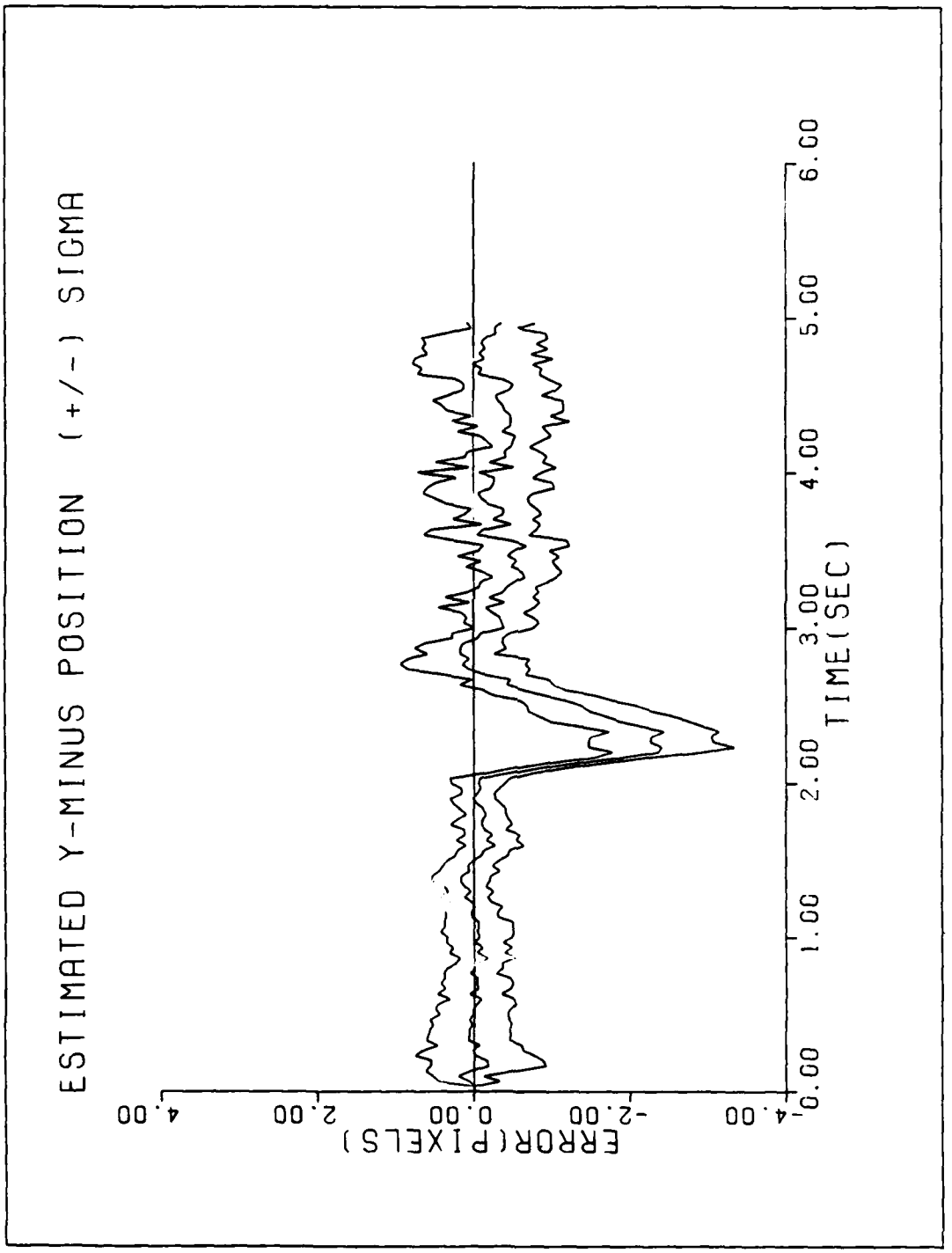


Figure C-4d. Performance Plot for T2G10MM-MAP (Retuned)

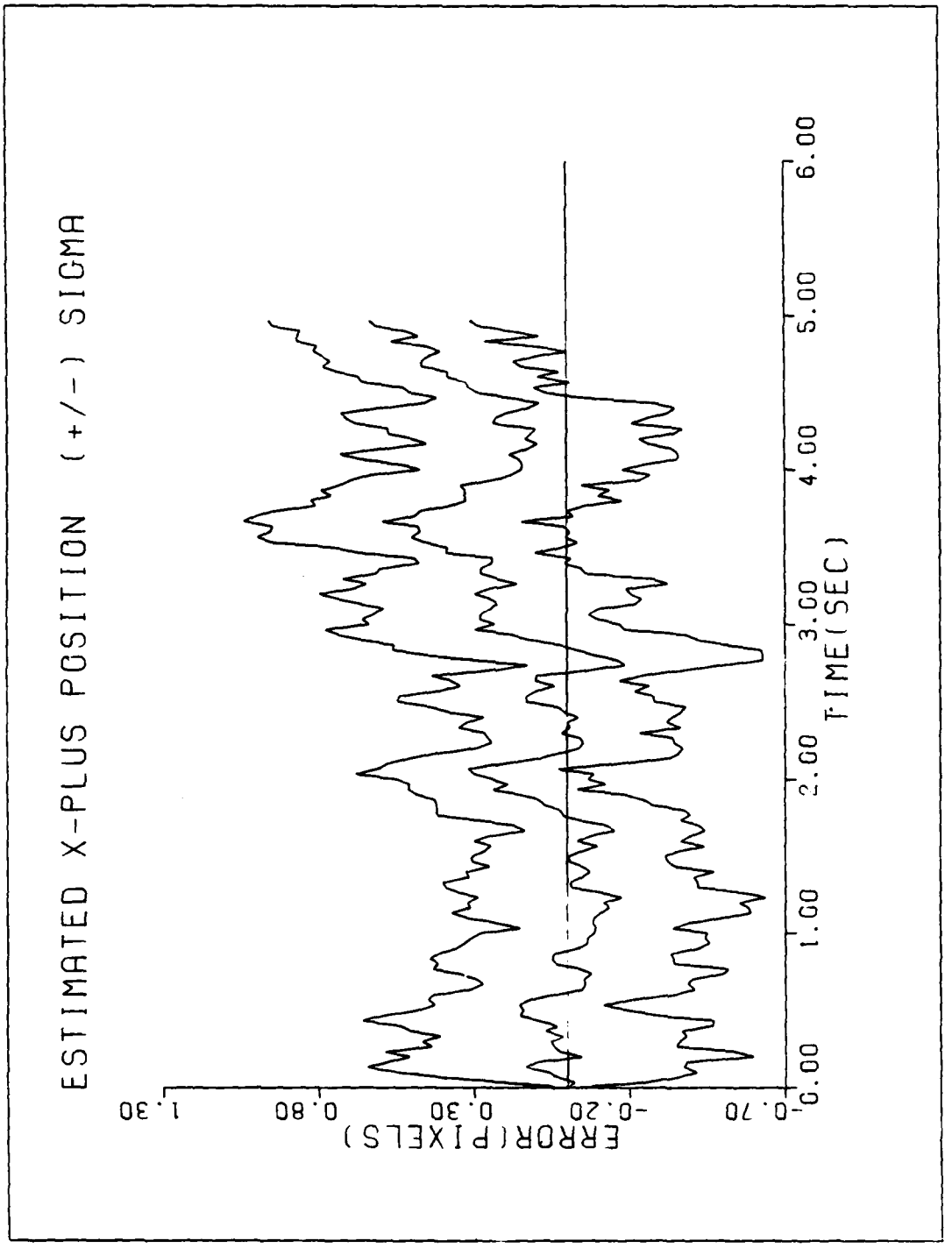


Figure C-4e. Performance Plot for T2G10MM-MAP (Retuned)

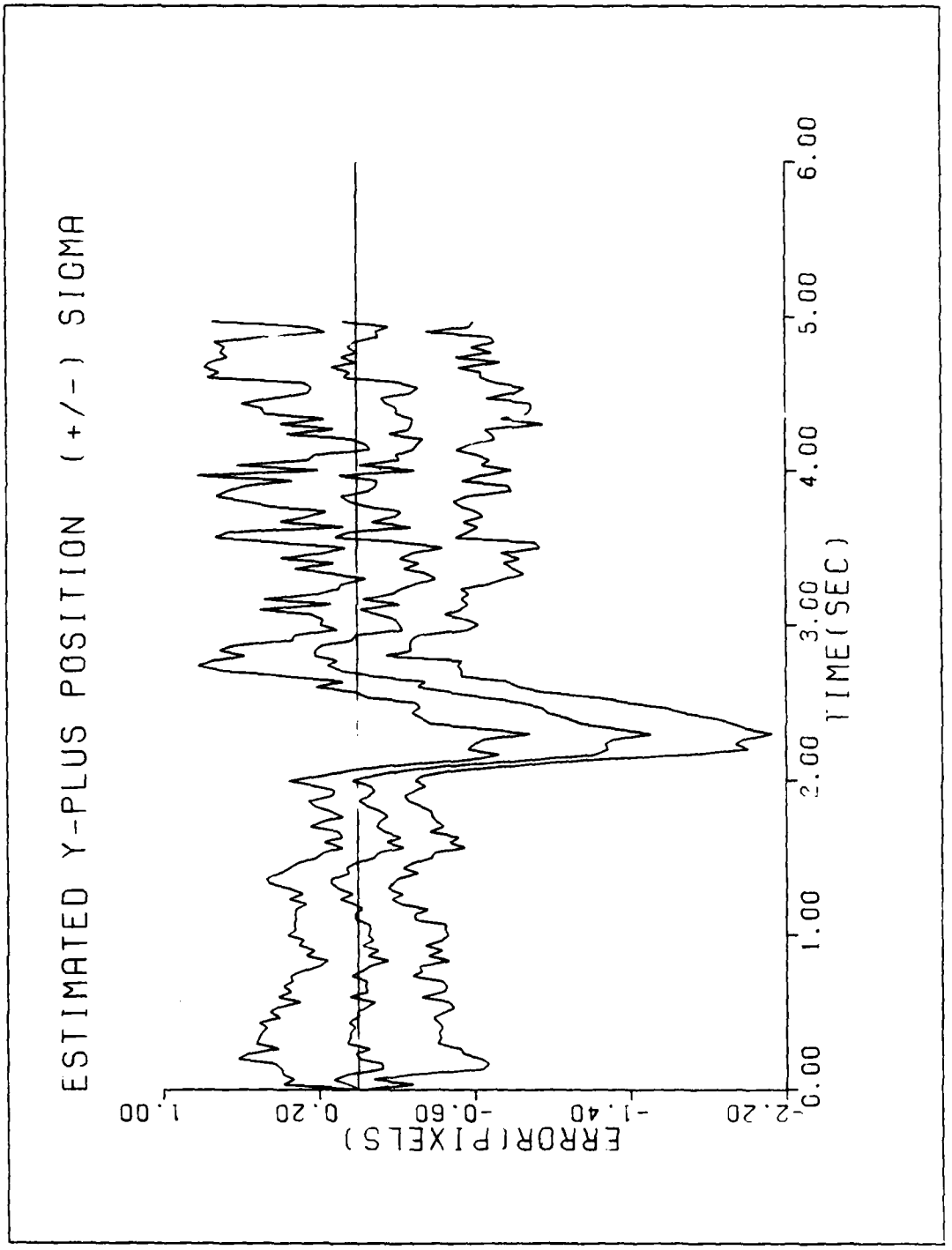


Figure C-4f. Performance Plot for T2G10MM-MAP (Retuned)

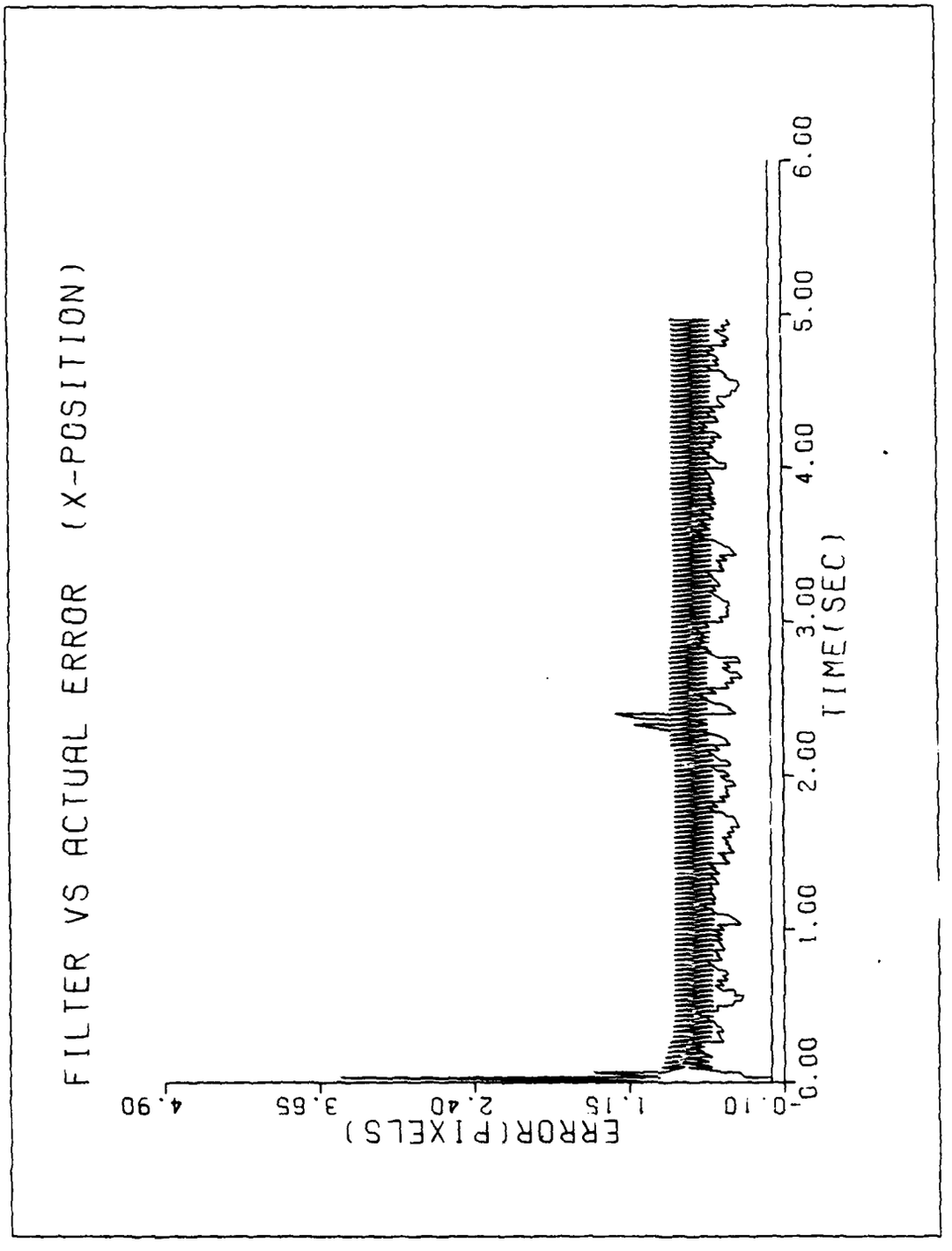


Figure C-5a. Performance Plot for T2G20MM-MAP

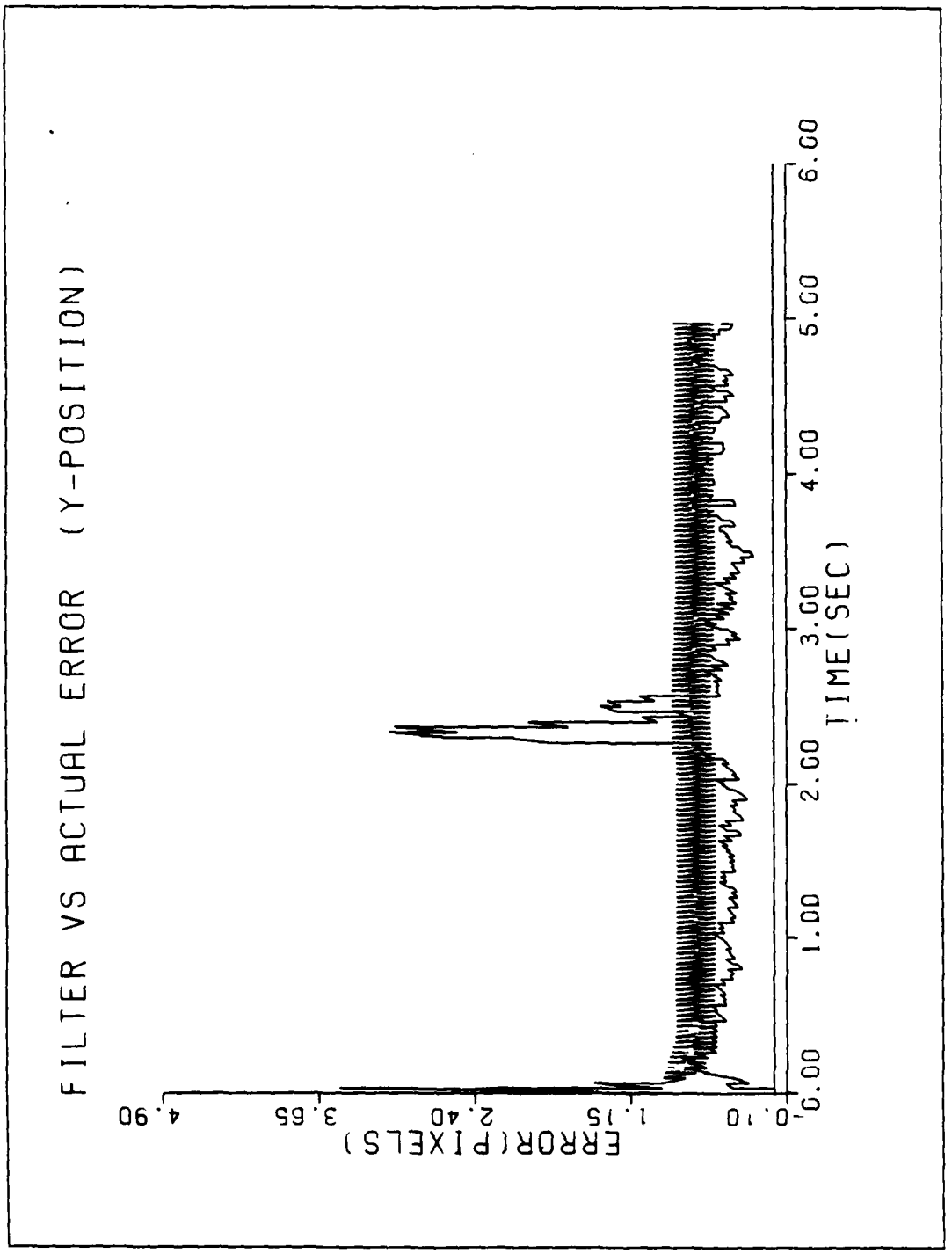


Figure C-5b. Performance Plot for T2G20MM-MAP

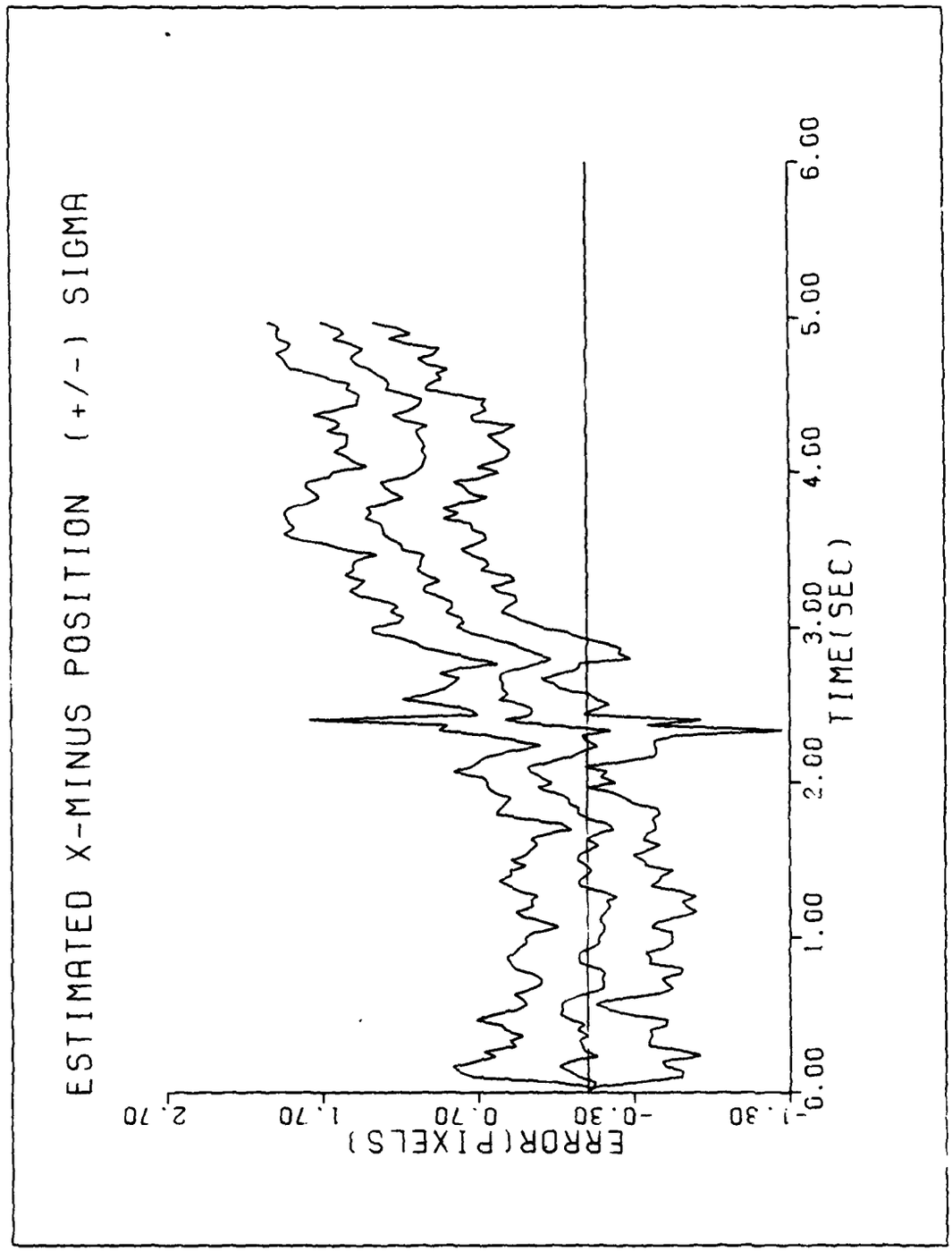


Figure C-5c. Performance Plot for T2G20MM-MAP

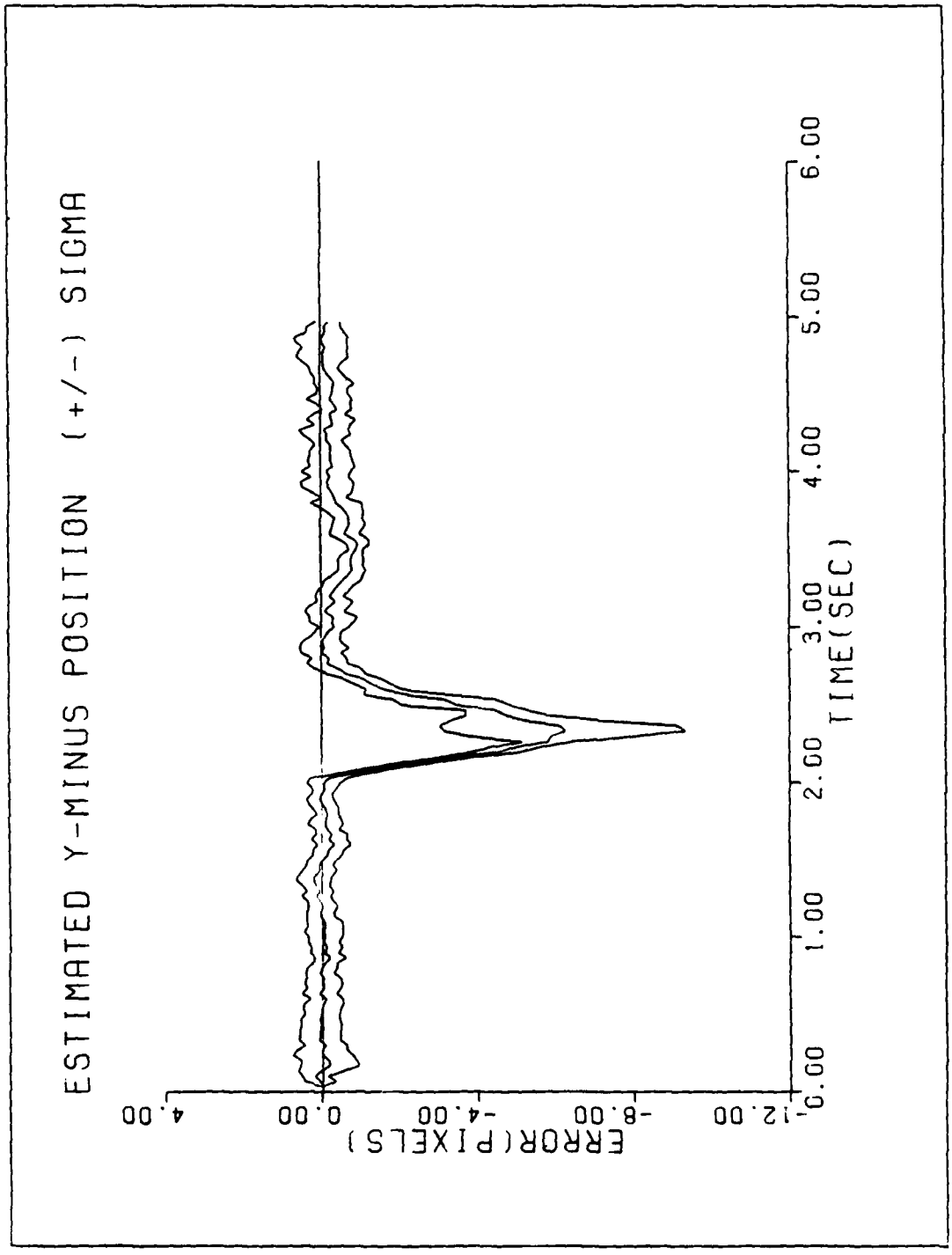


Figure C-5d. Performance Plot for T2G20MM-MAP

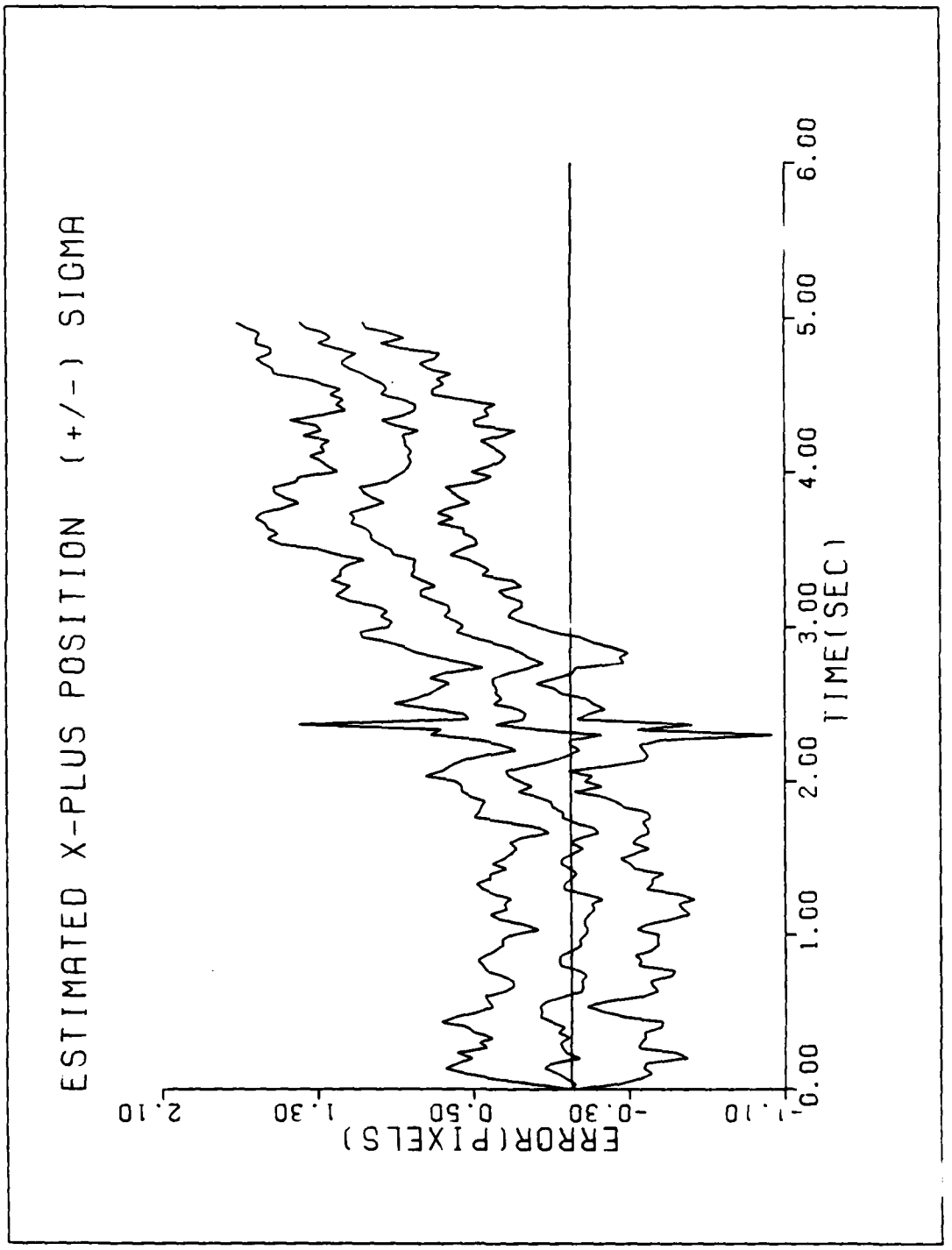


Figure C-5e. Performance Plot for T2G20MM-MAP

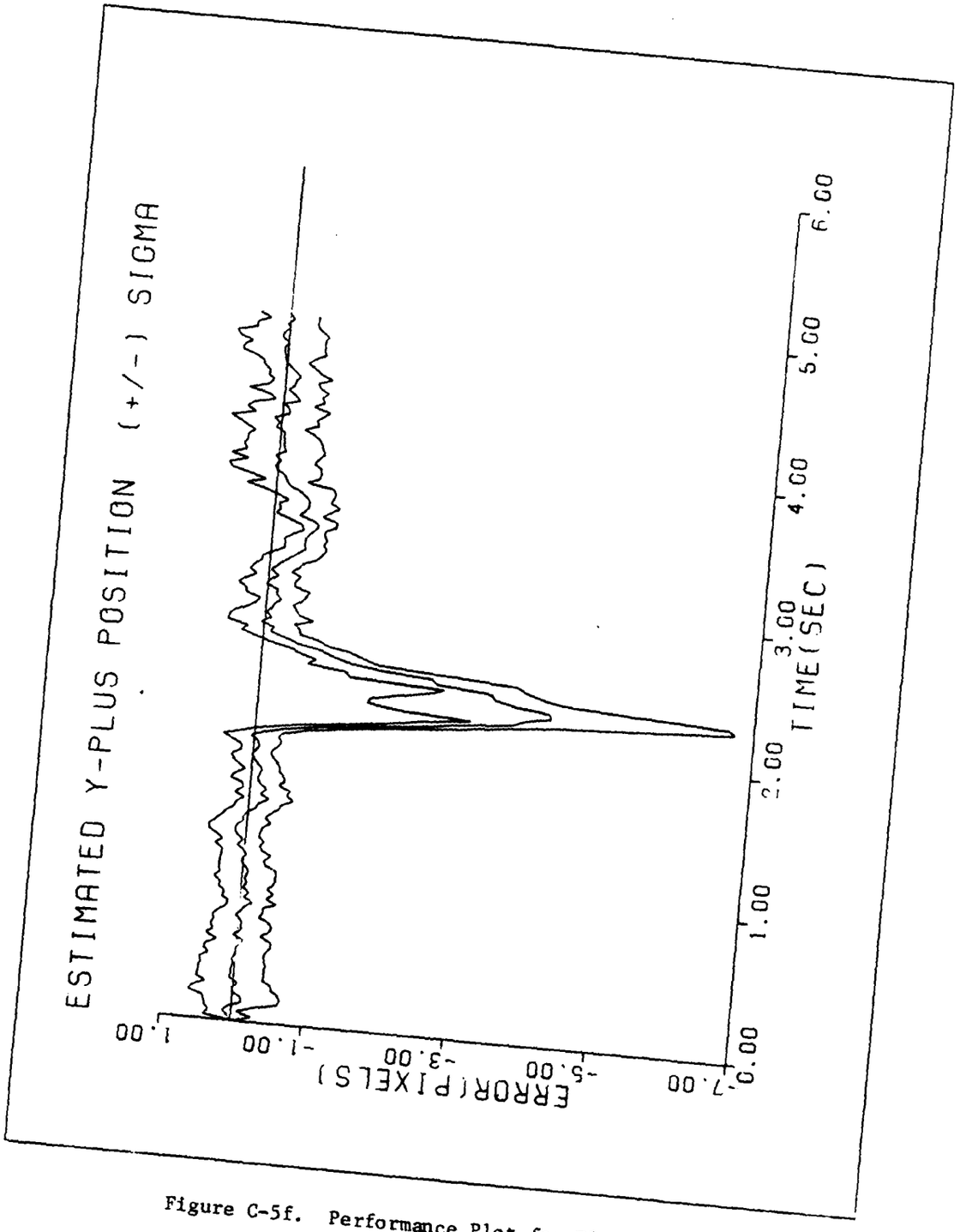


Figure C-5f. Performance Plot for T2G20MM-MAP

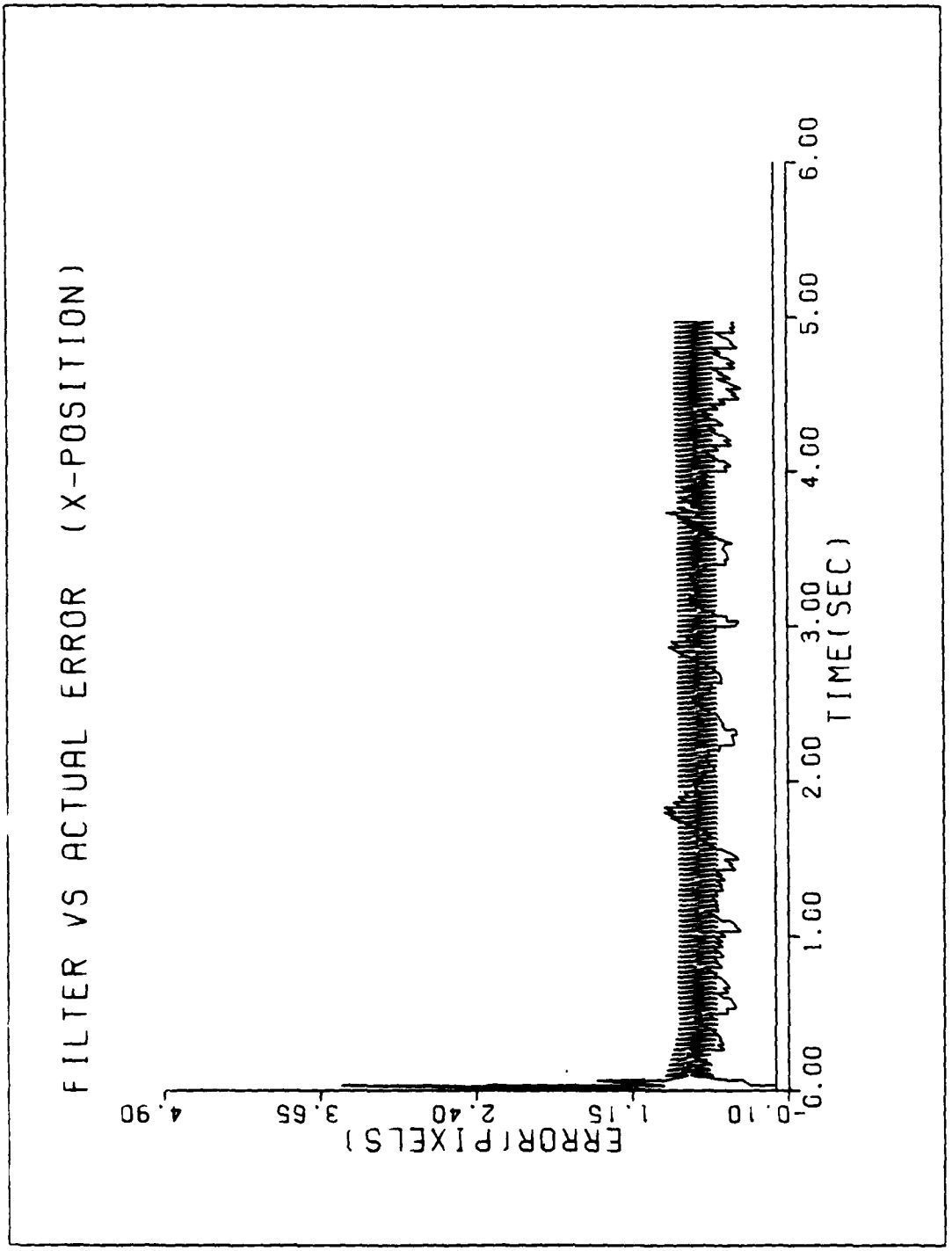


Figure C-6a. Performance Plot for T3G2MM-MAP

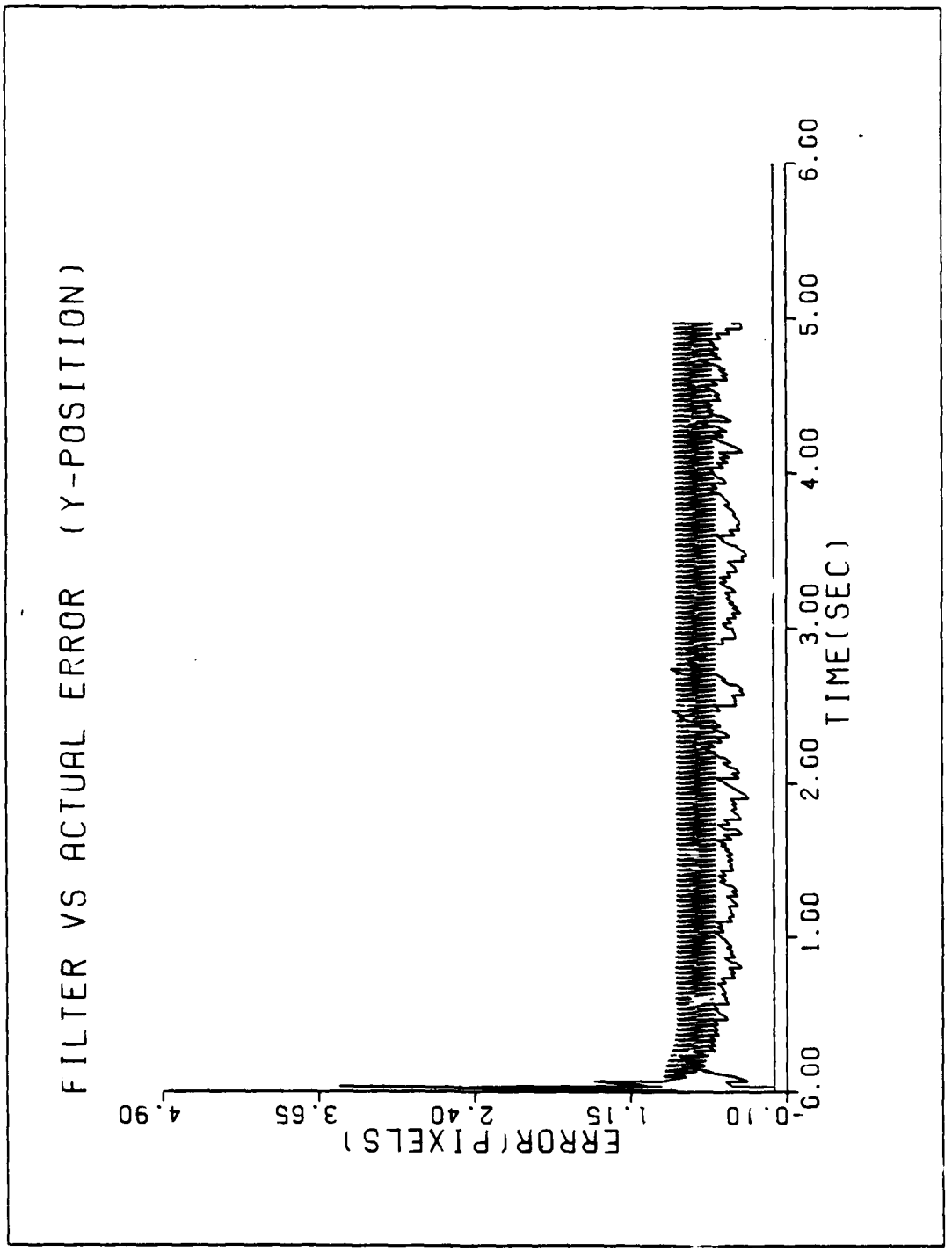


Figure C-6b. Performance Plot for T3G2MM-MAP

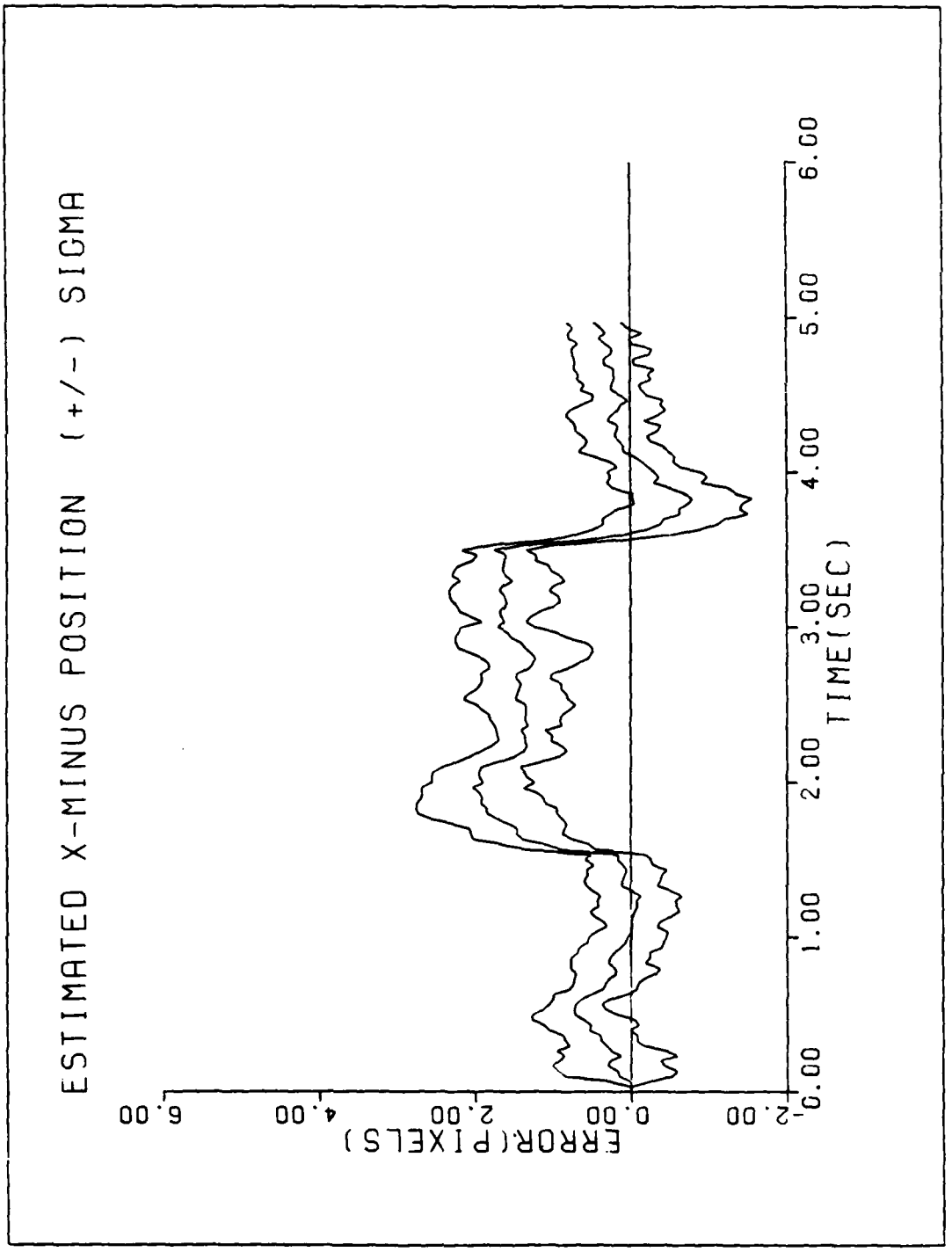


Figure C-6c. Performance Plot for T3G2MM-MAP

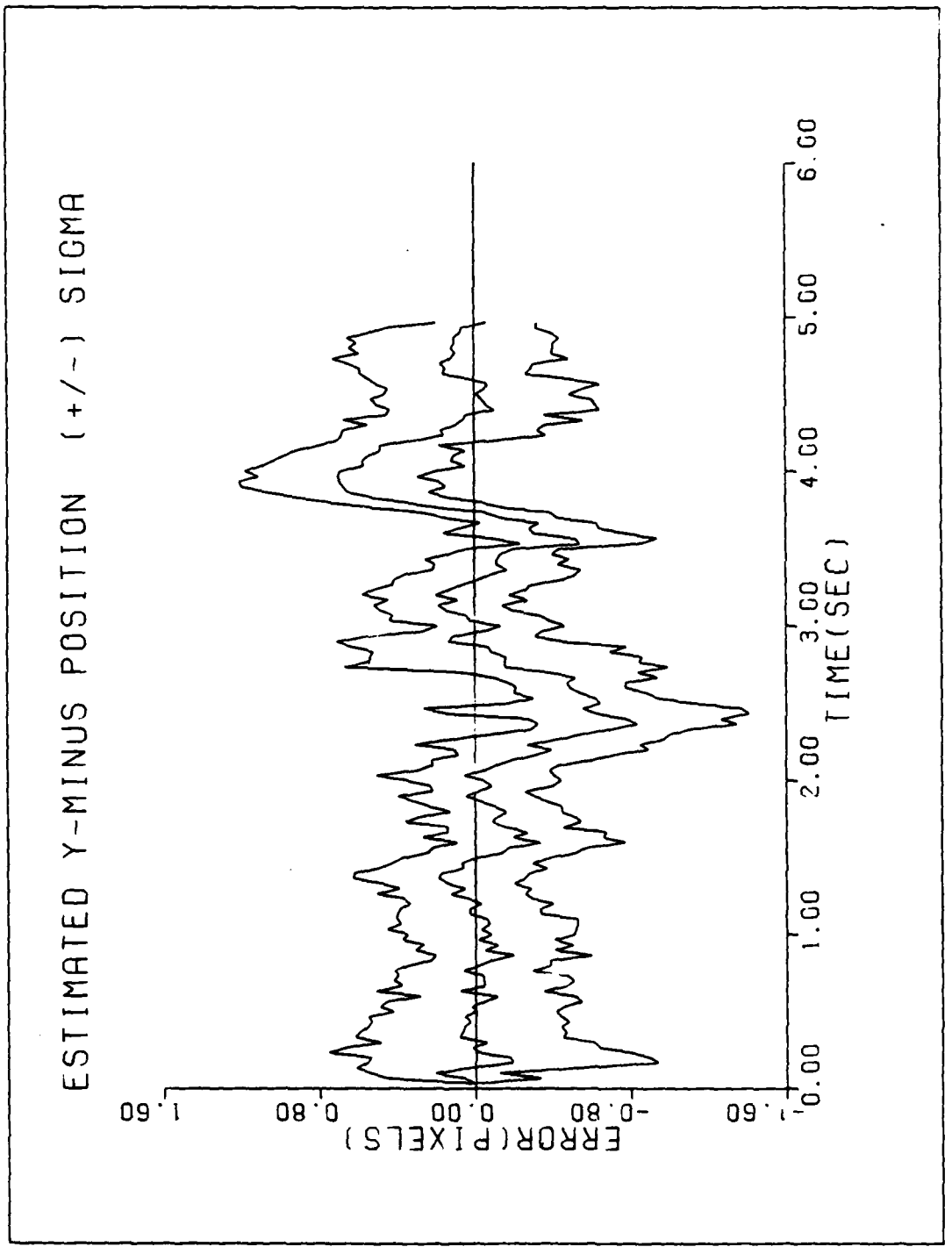


Figure C-6d. Performance Plot for T3G2MM-MAP

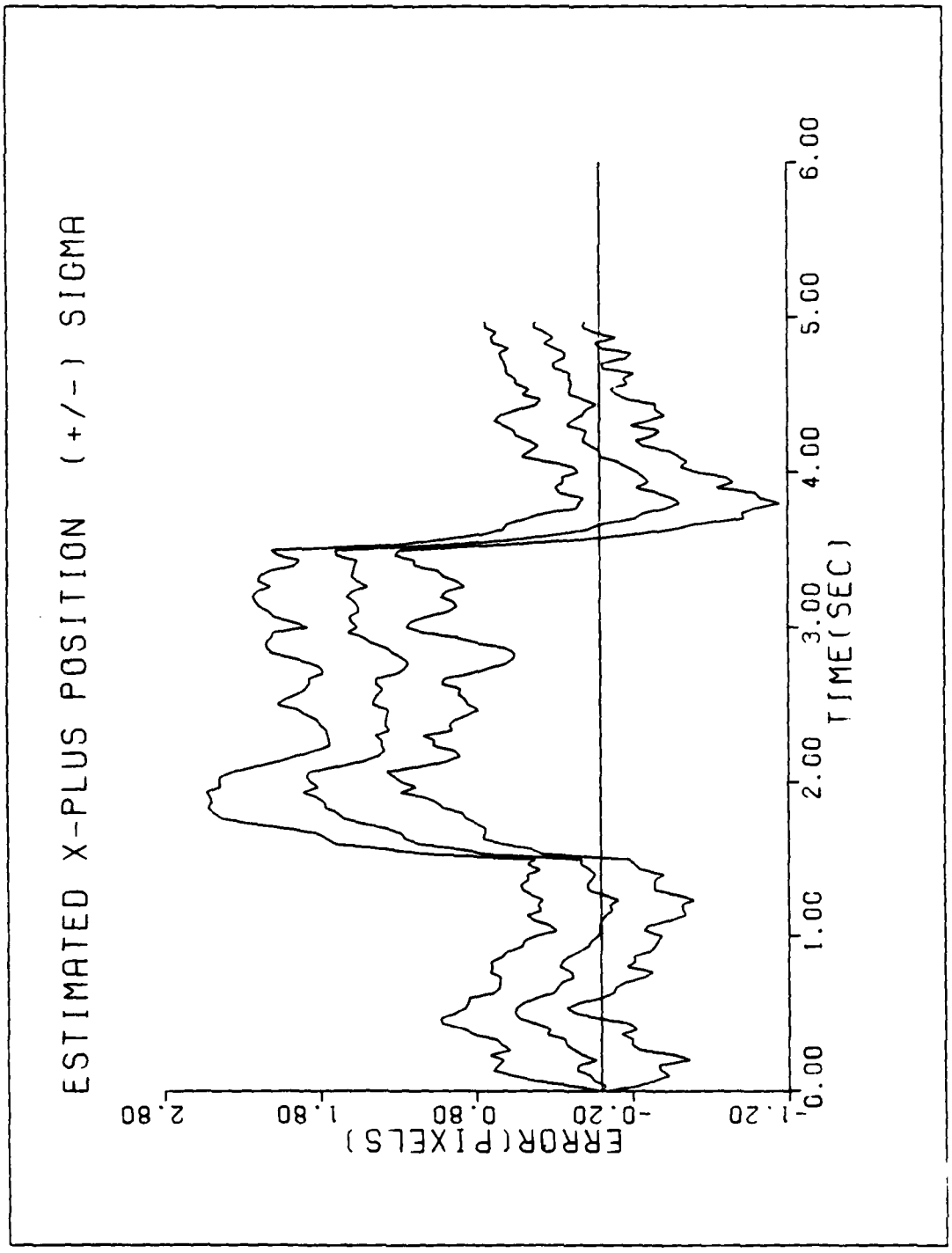


Figure C-6e. Performance Plot for T3G2MM-MAP

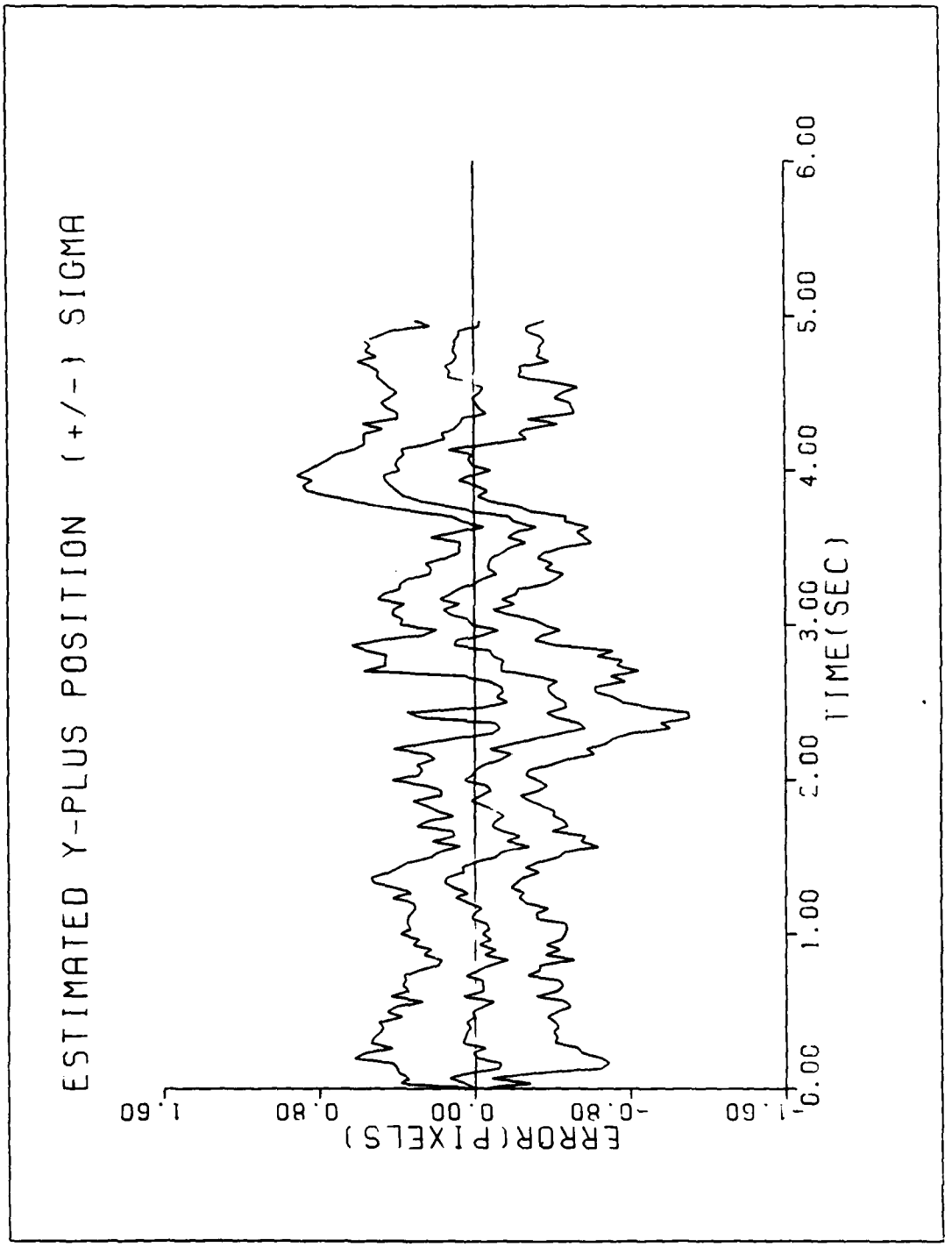


Figure C-6f. Performance Plot for T3G2MM-MAP

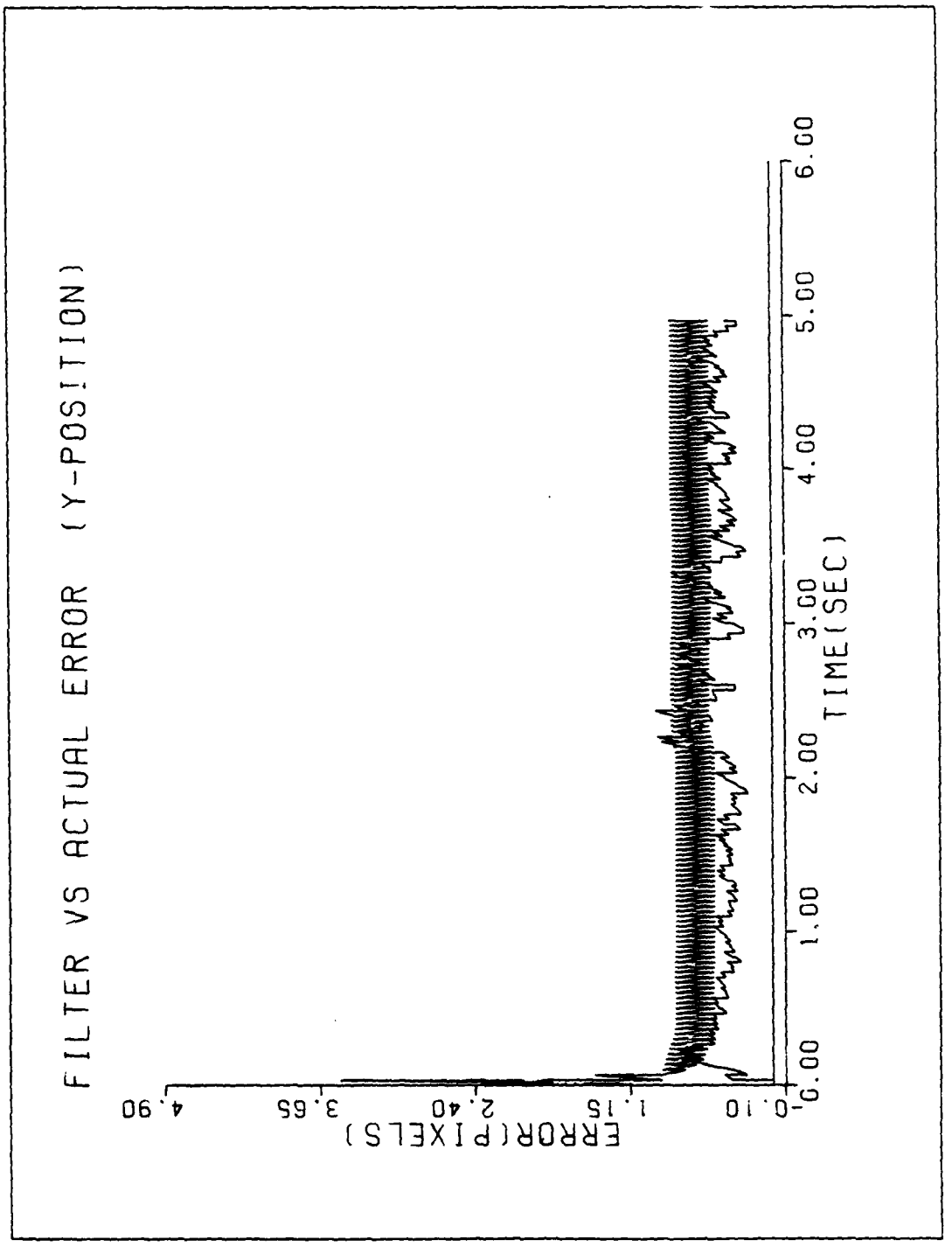


Figure C-7a. Performance Plot for T3G10MM-MAP

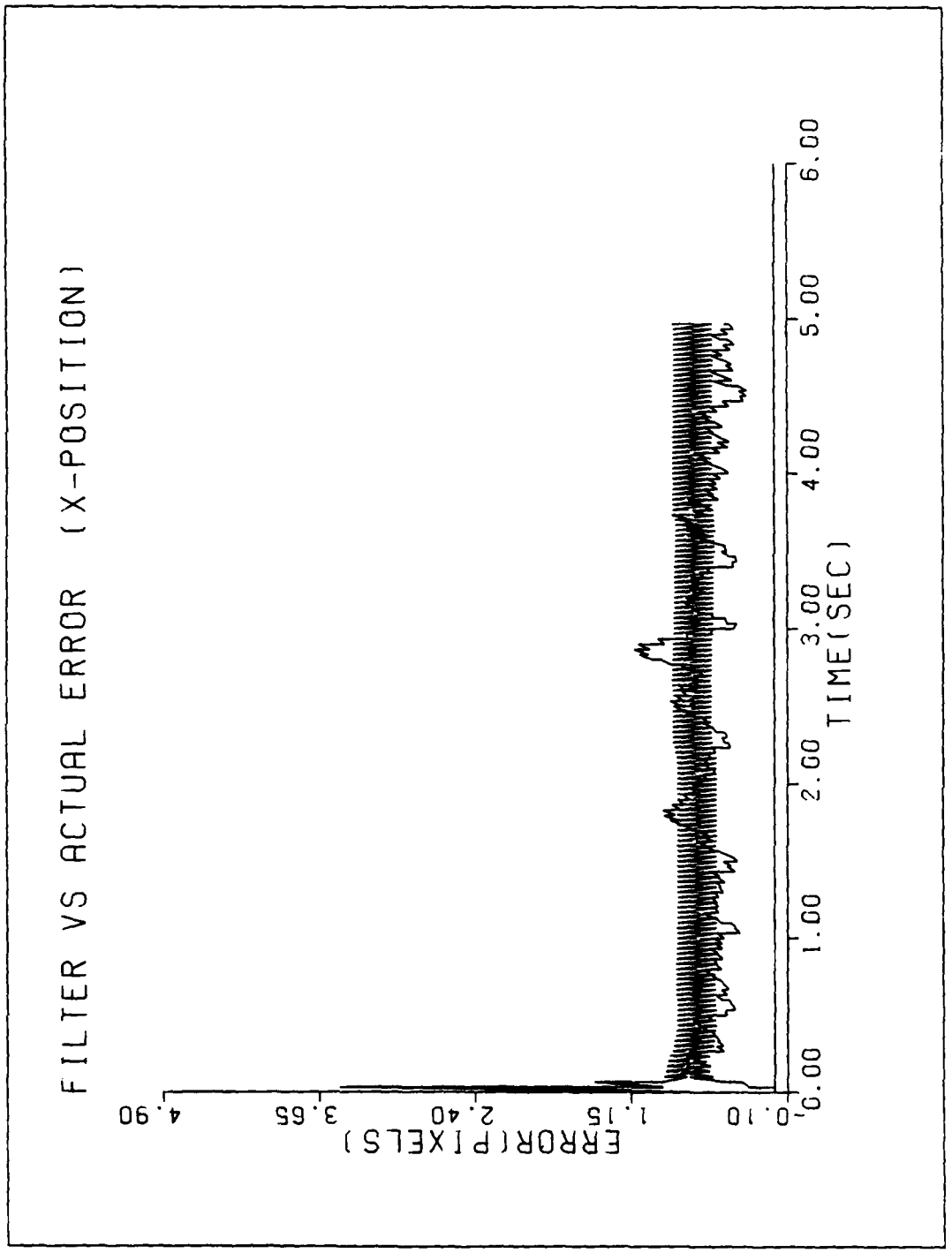


Figure C-7b. Performance Plot for T3G10MM-MAP

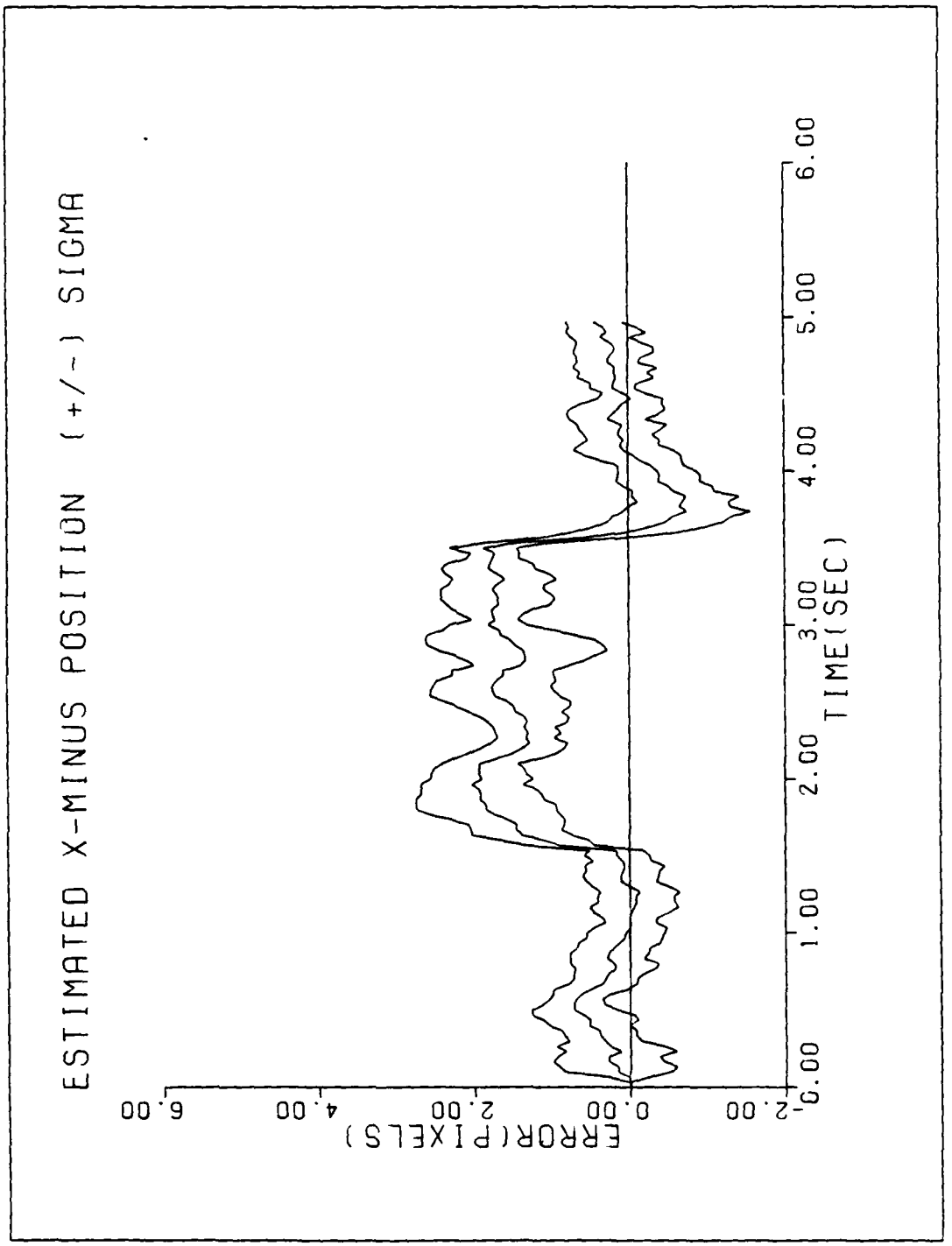


Figure C-7c. Performance Plot for T3G10MM-MAP

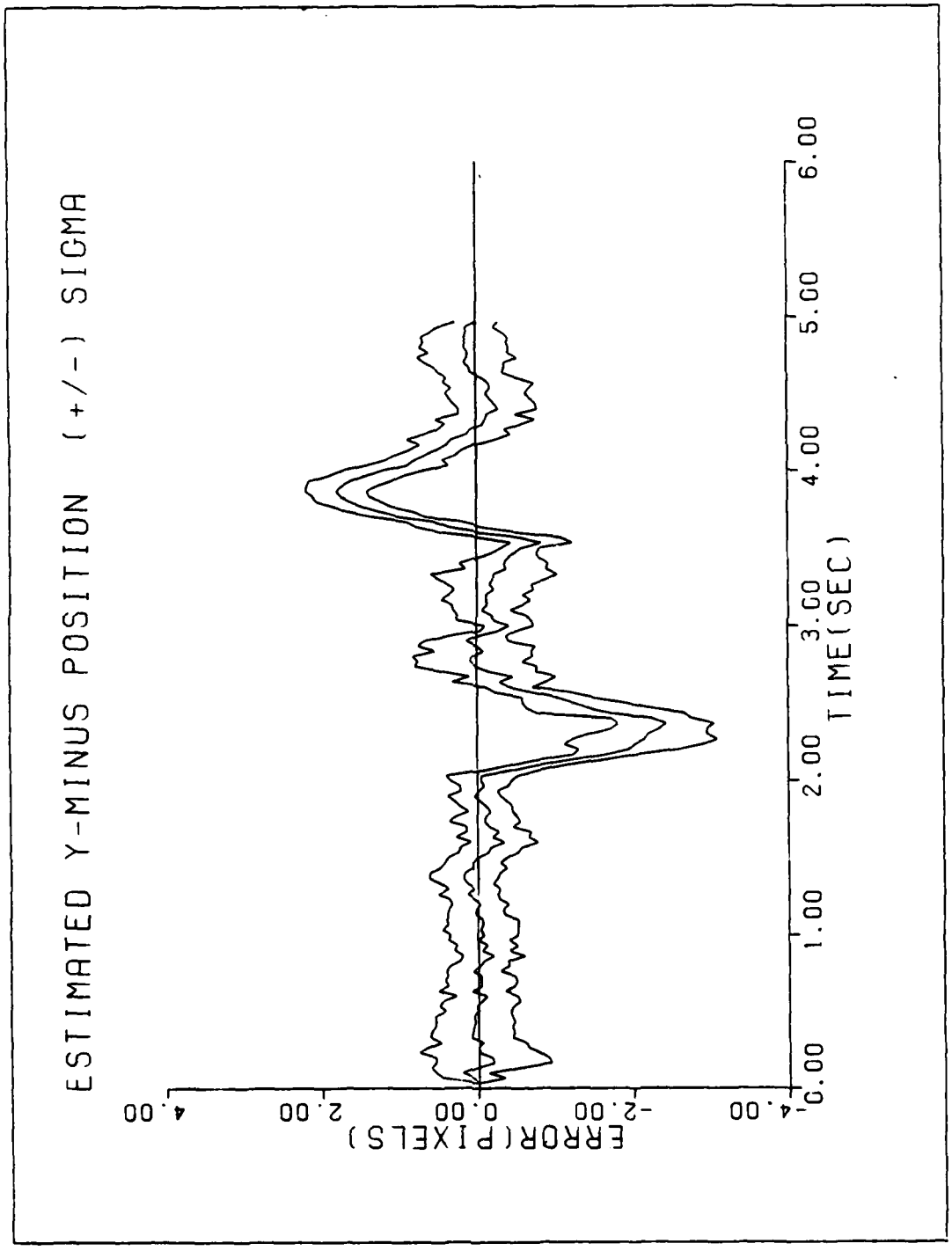


Figure C-7d. Performance Plot for T3G10MM-MAP

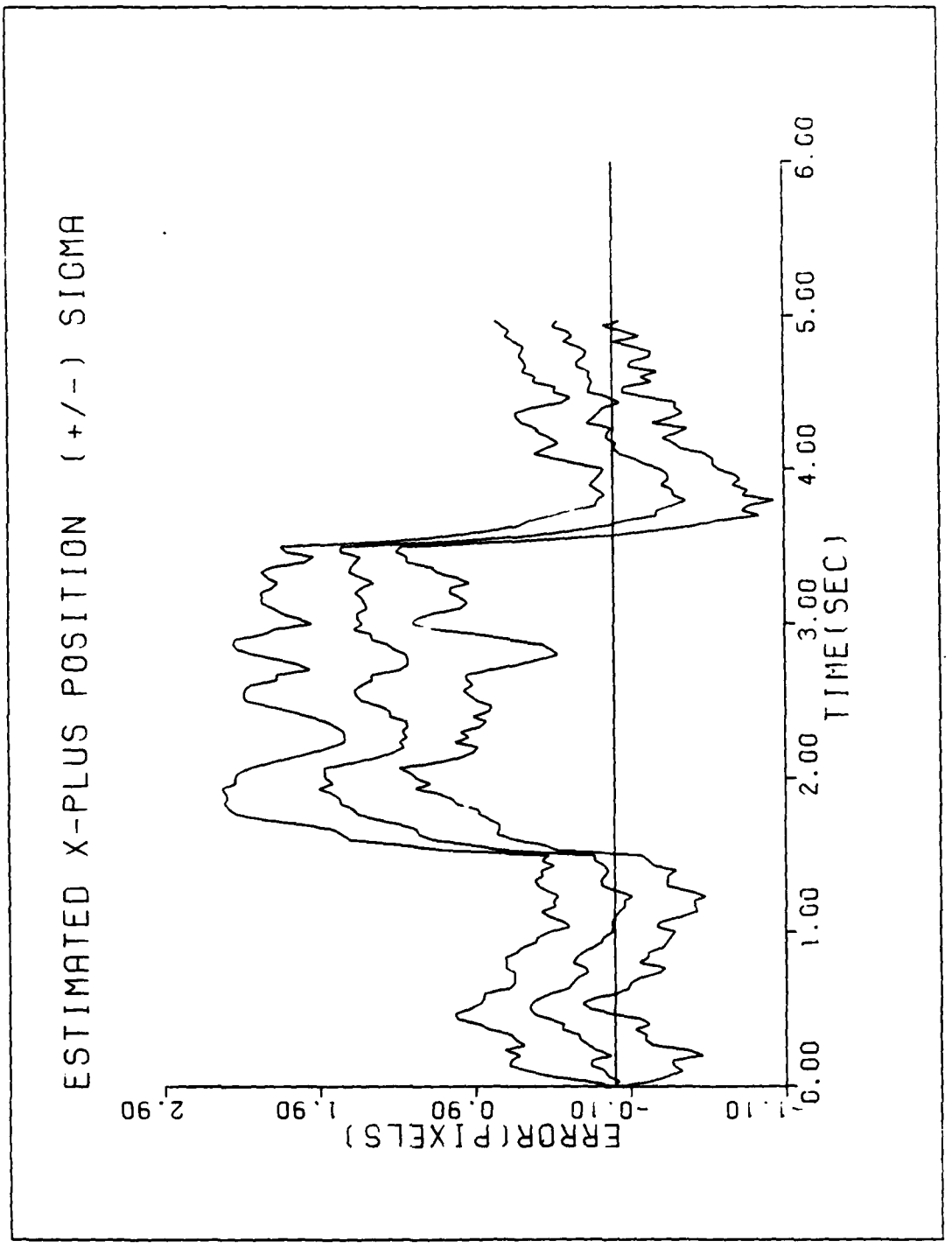


Figure C-7e. Performance Plot for T3G10MM-MAP

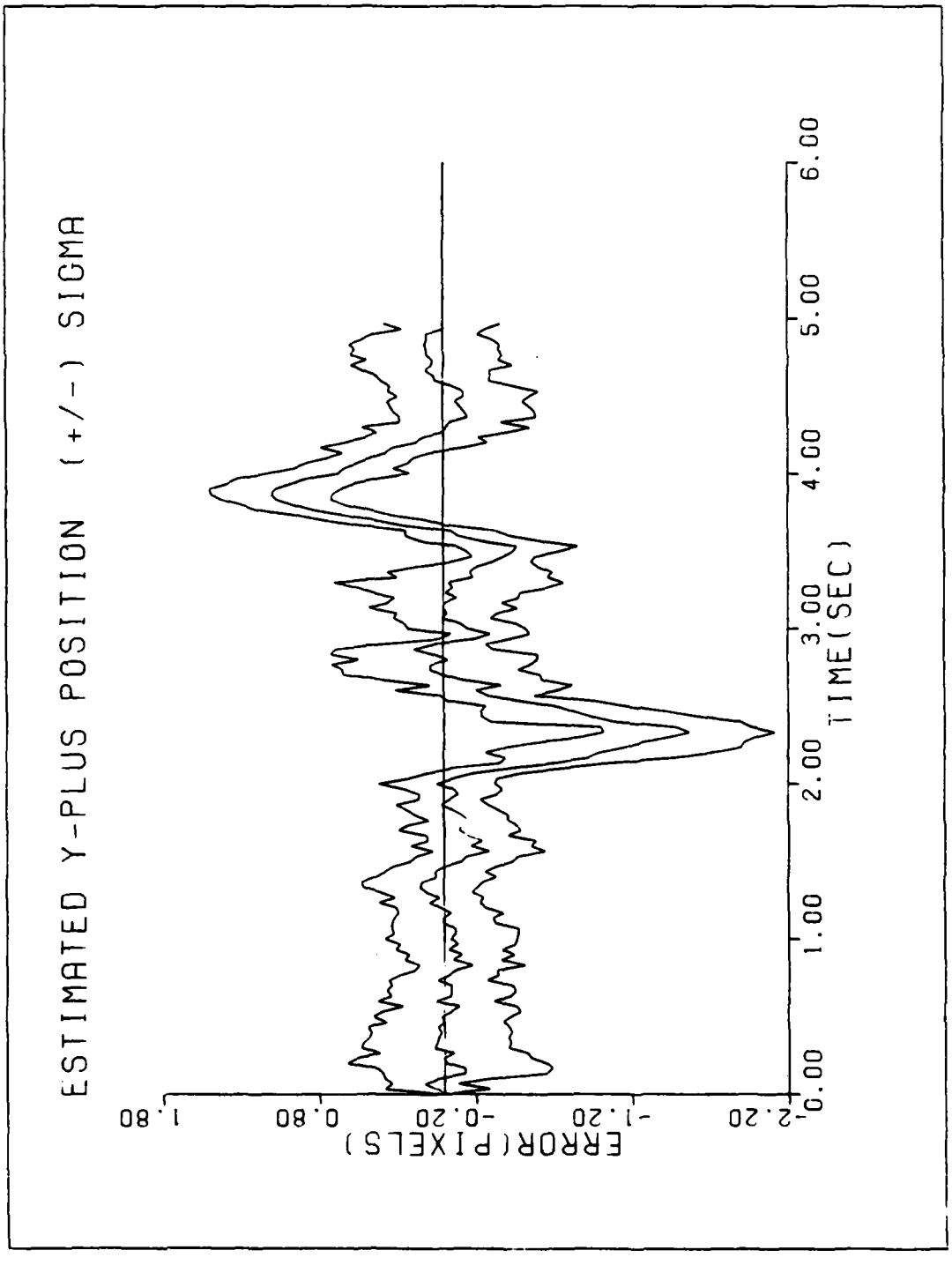


Figure C-7f. Performance Plot for T3G10MM-MAP

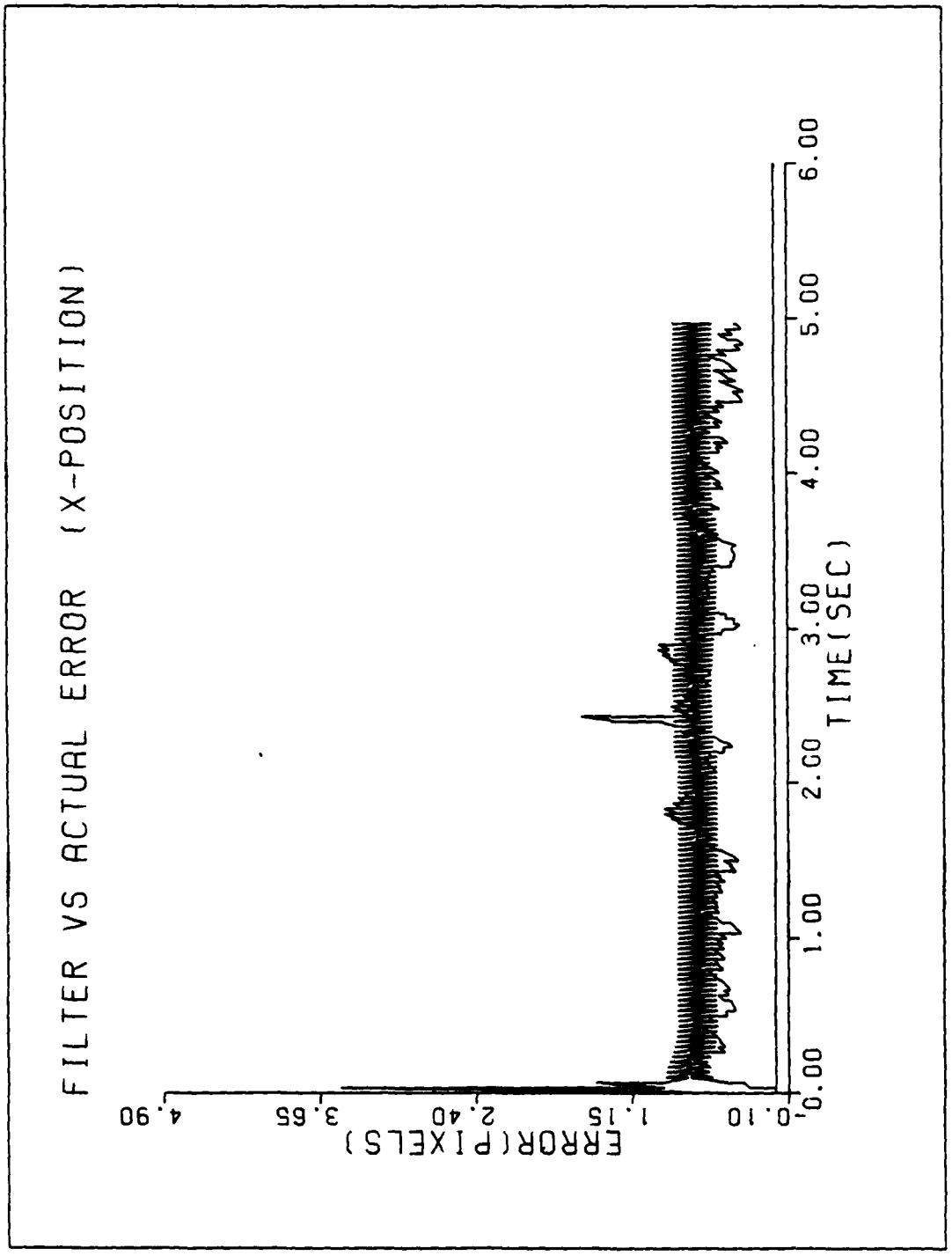


Figure C-8a. Performance Plot for T3G20MM-MAP

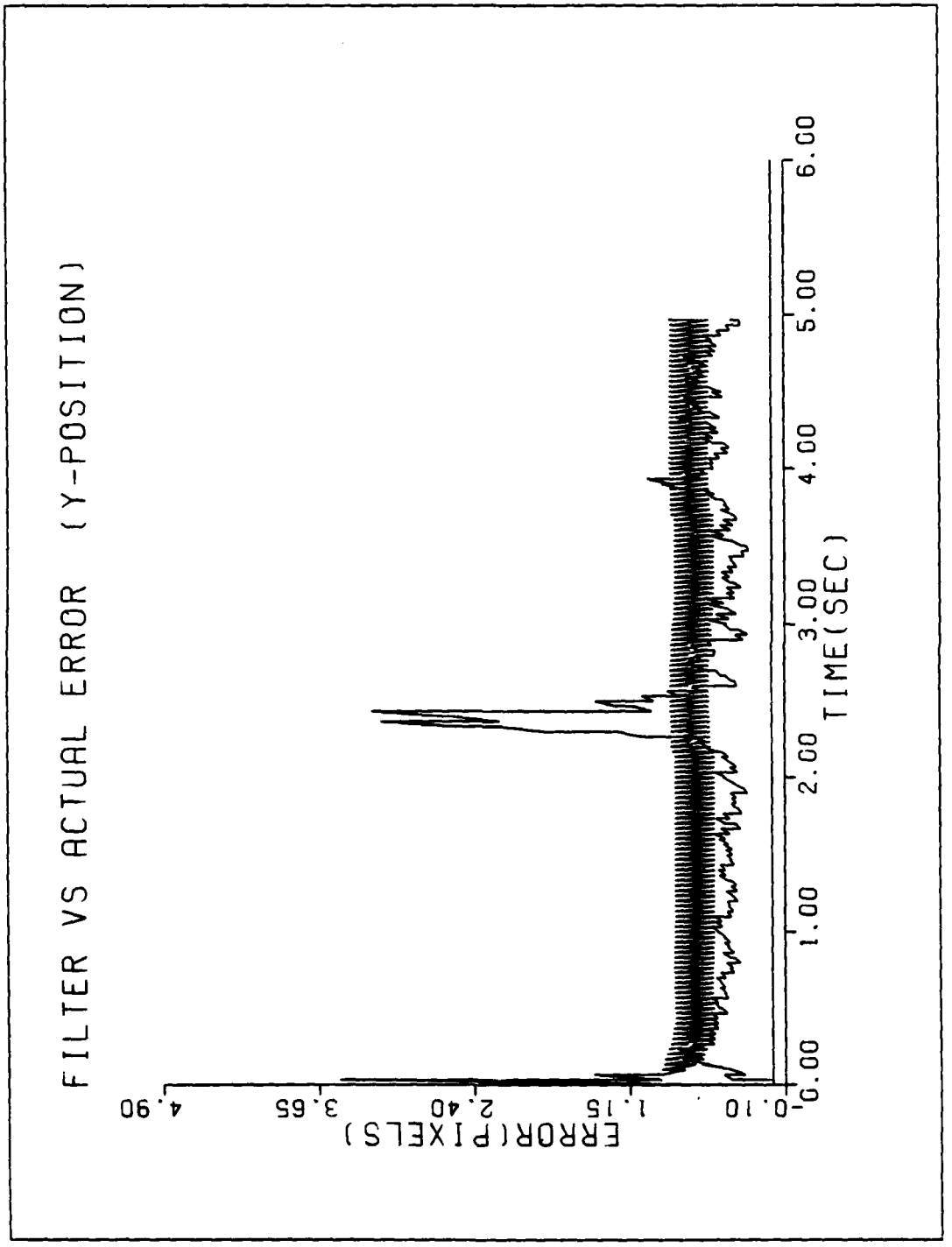


Figure C-8b. Performance Plot for T3G20MM-MAP

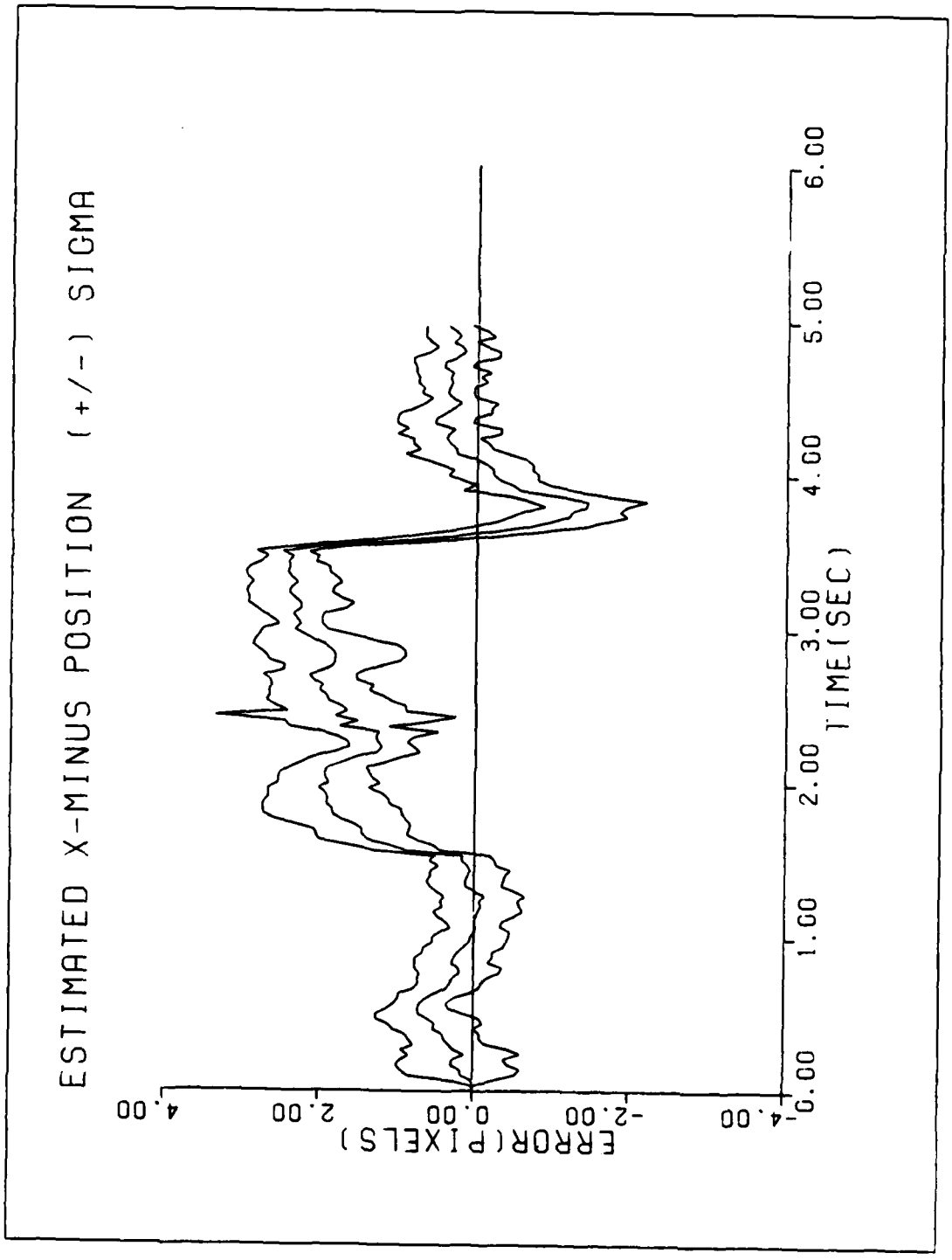


Figure C-8c. Performance Plot for T3G20MM-MAP

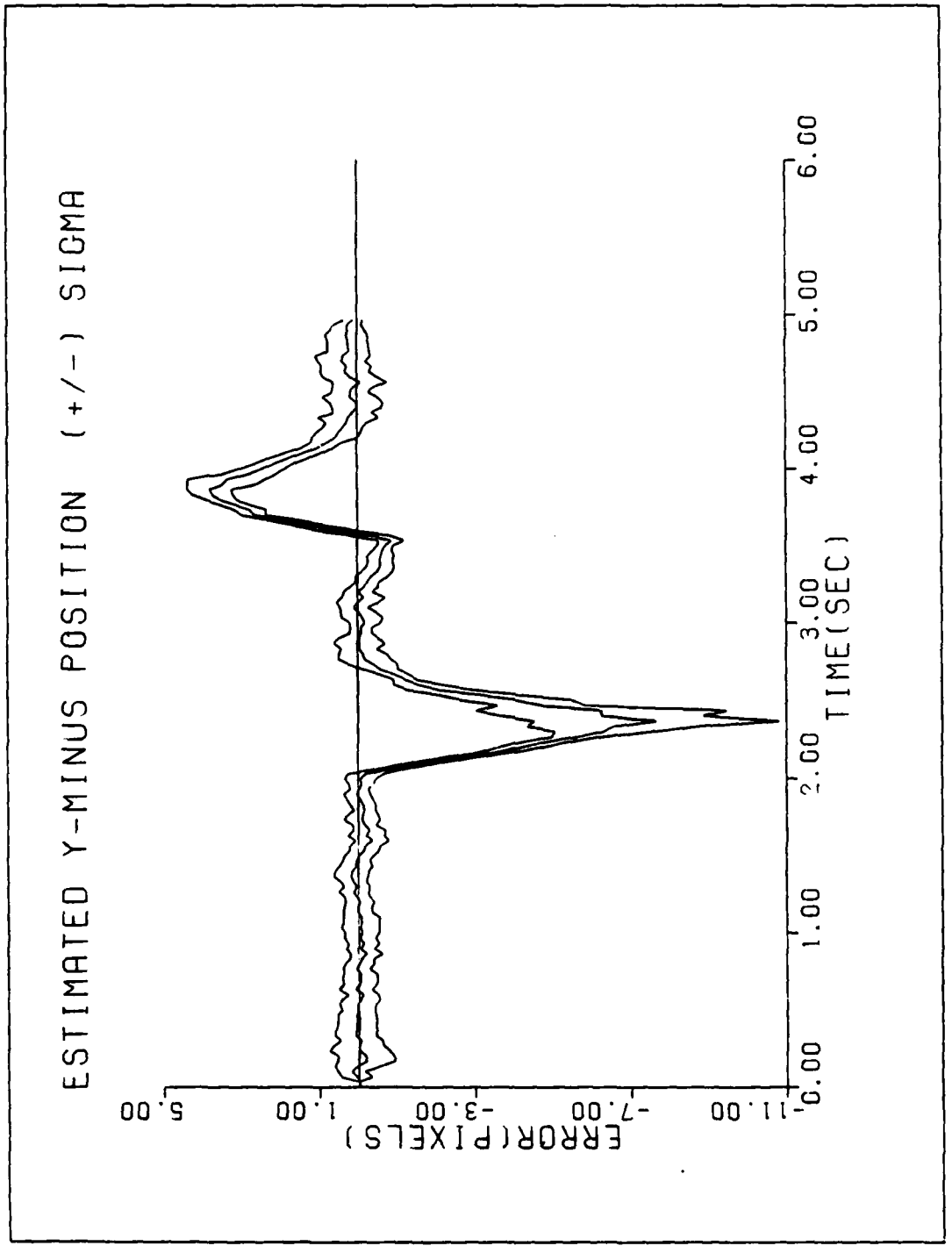


Figure C-8d. Performance Plot for T3G20MM-MAP

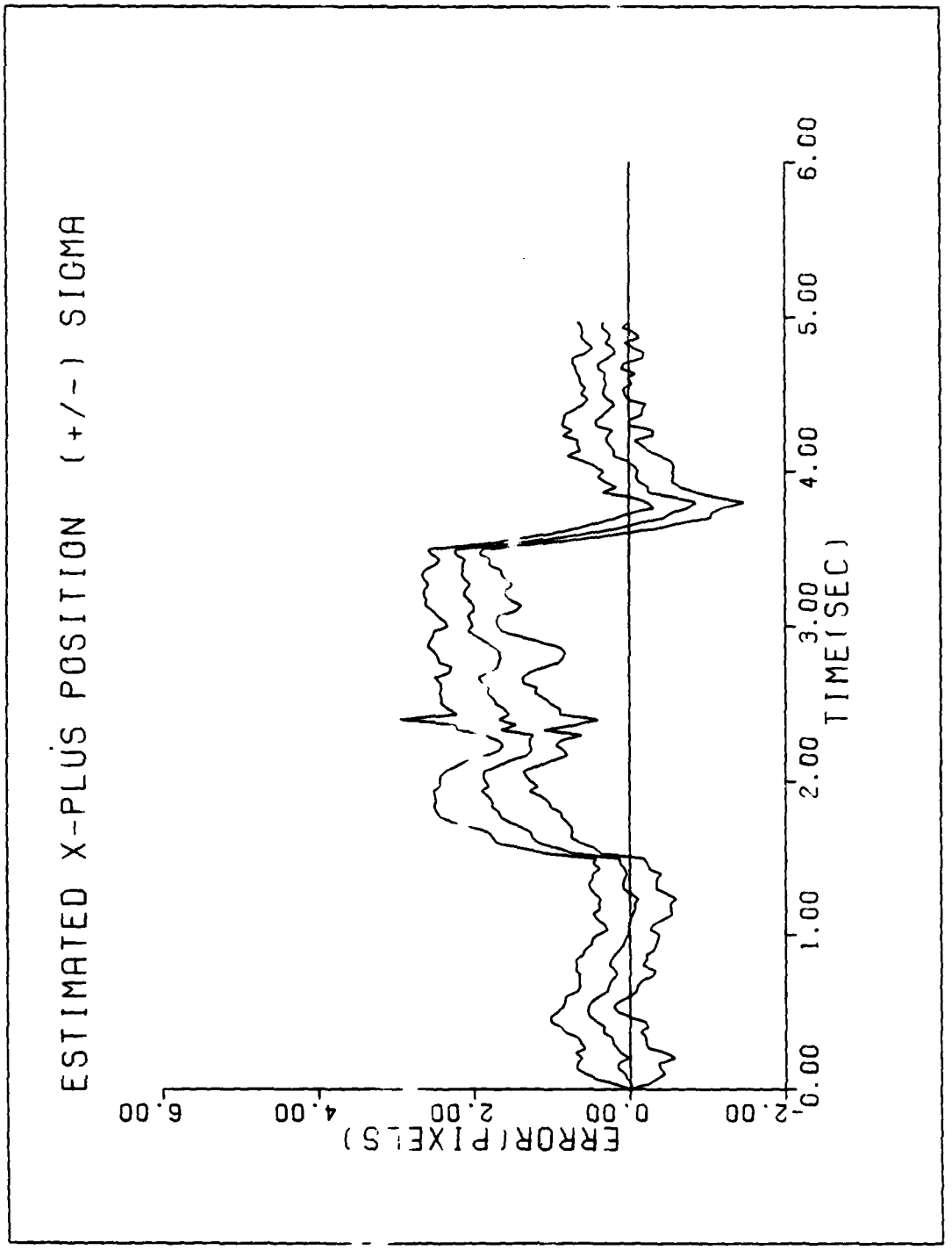


Figure C-8e. Performance Plot for T3G20MM-MAP

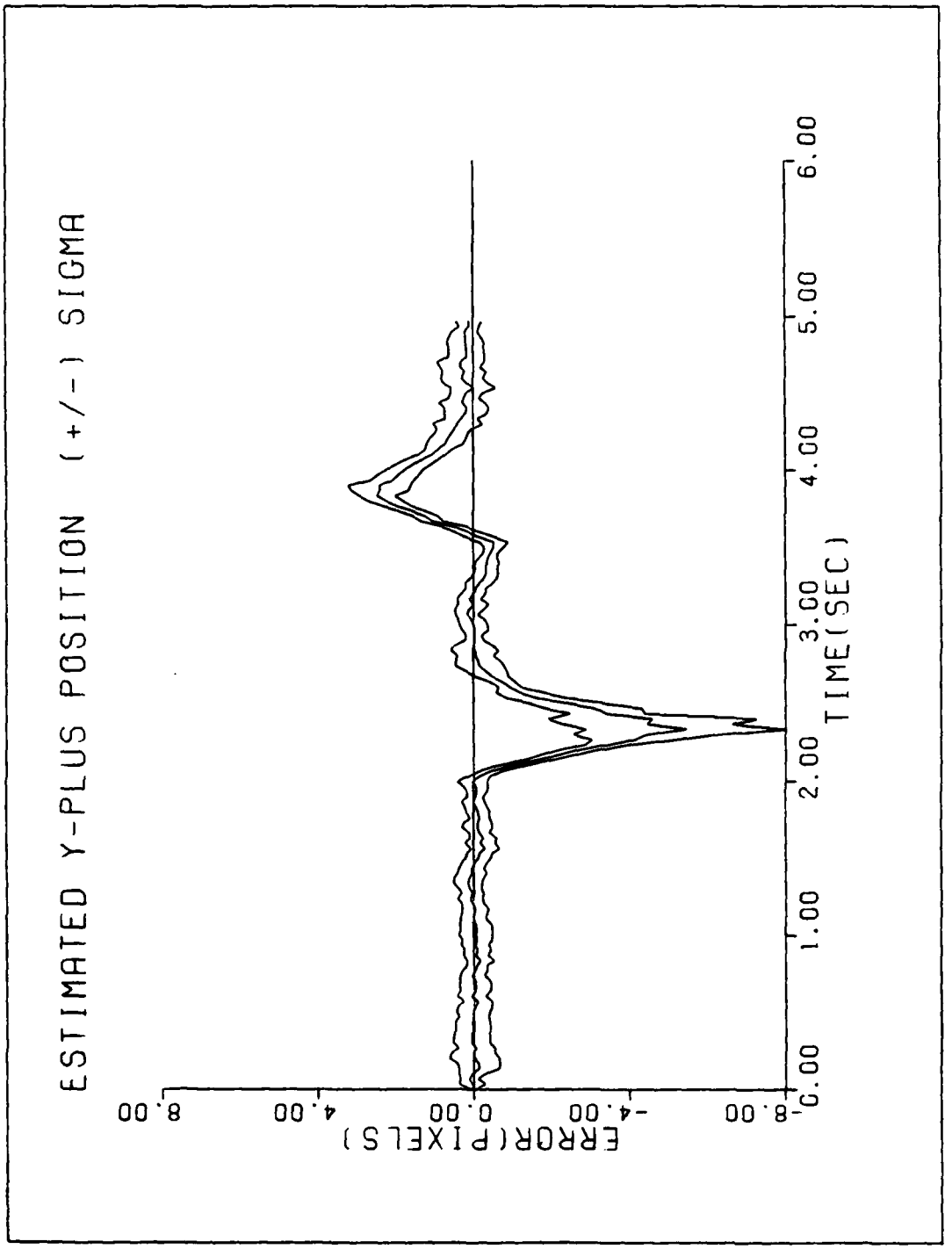


Figure C-8f. Performance Plot for T3G20MM-MAP

APPENDIX D

Performance Plots for the Baseline MMAF - Extended Run

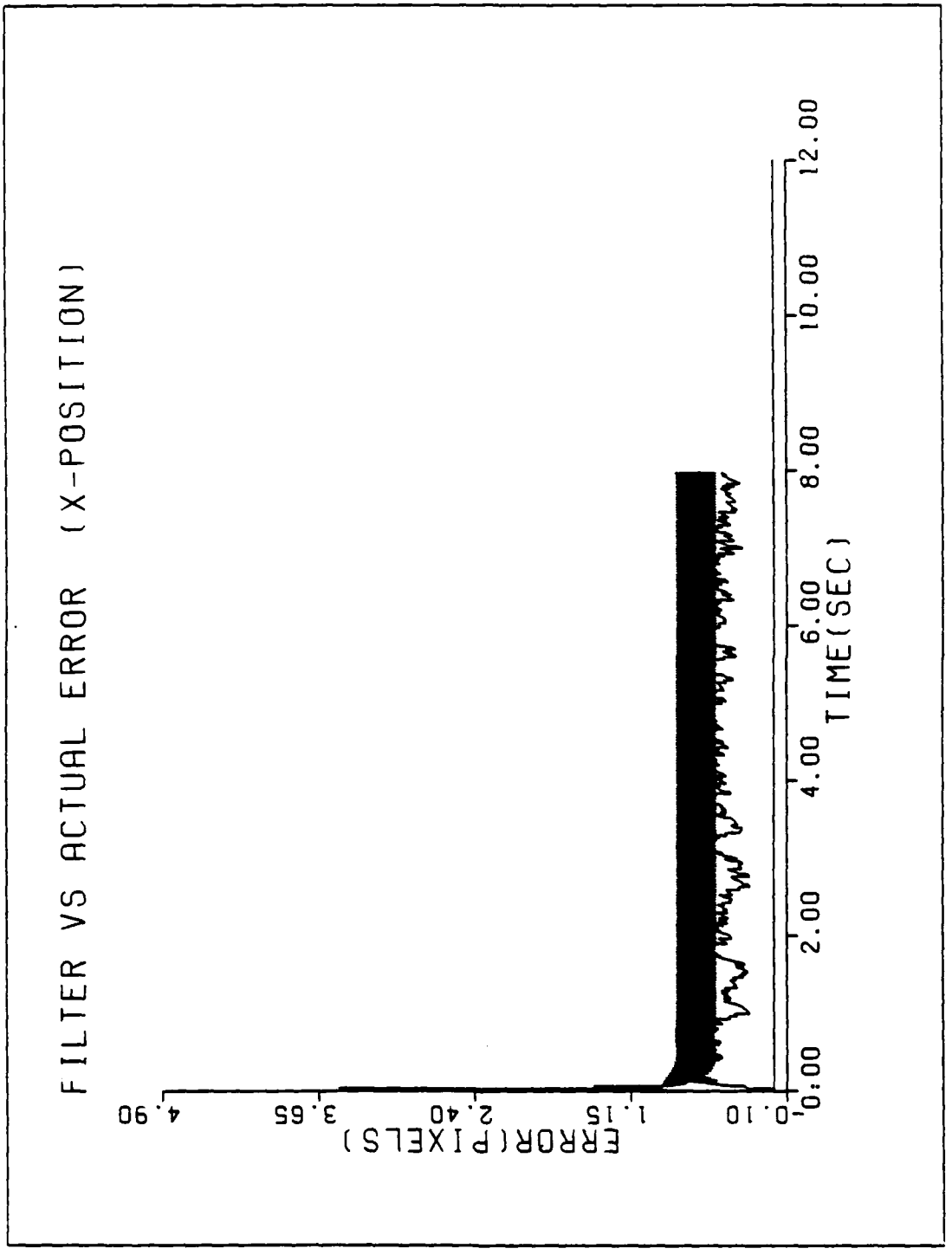


Figure D-1a. Performance Plot for TIM-XT

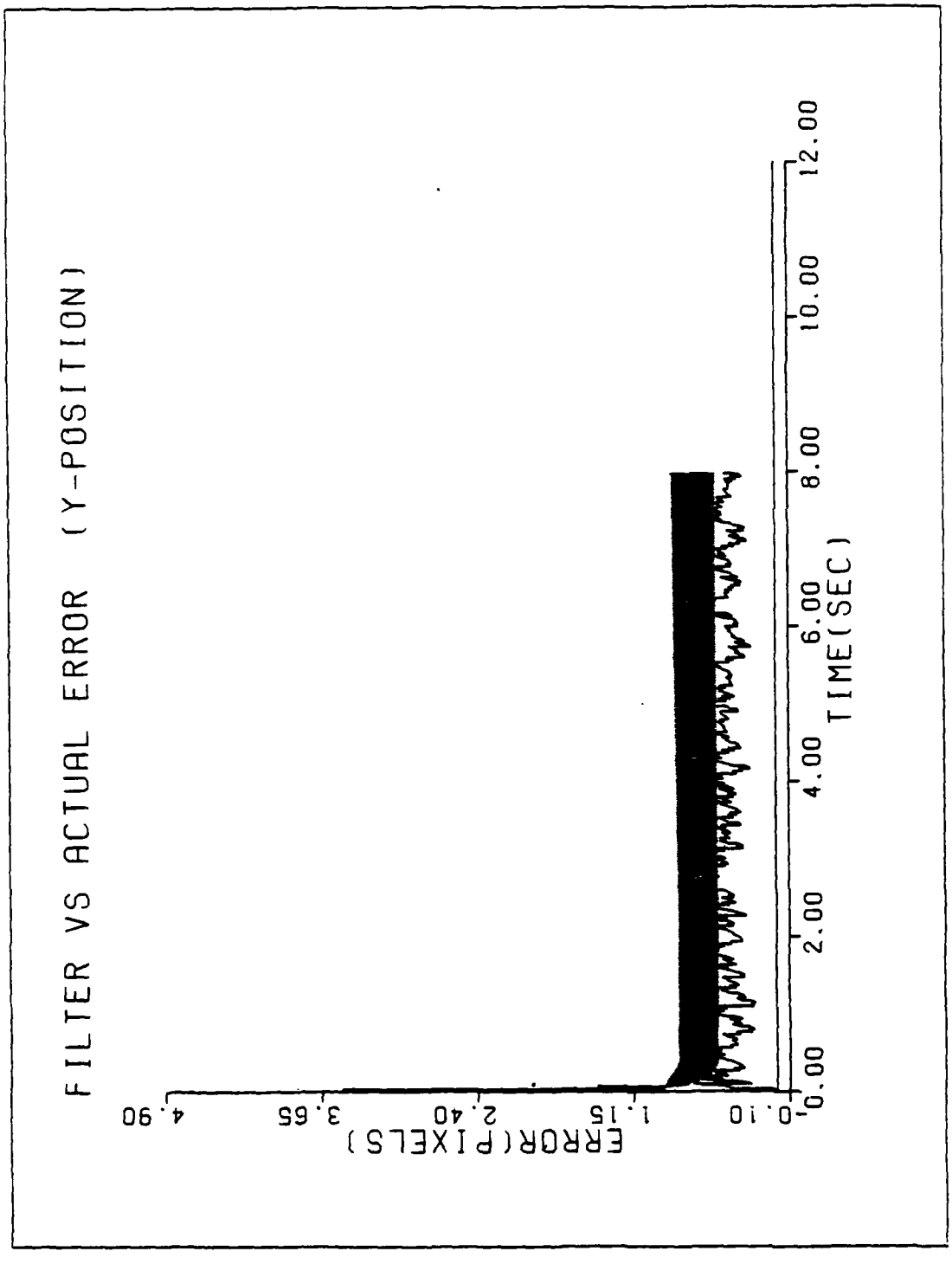


Figure D-1b. Performance Plot for TEEM-XT

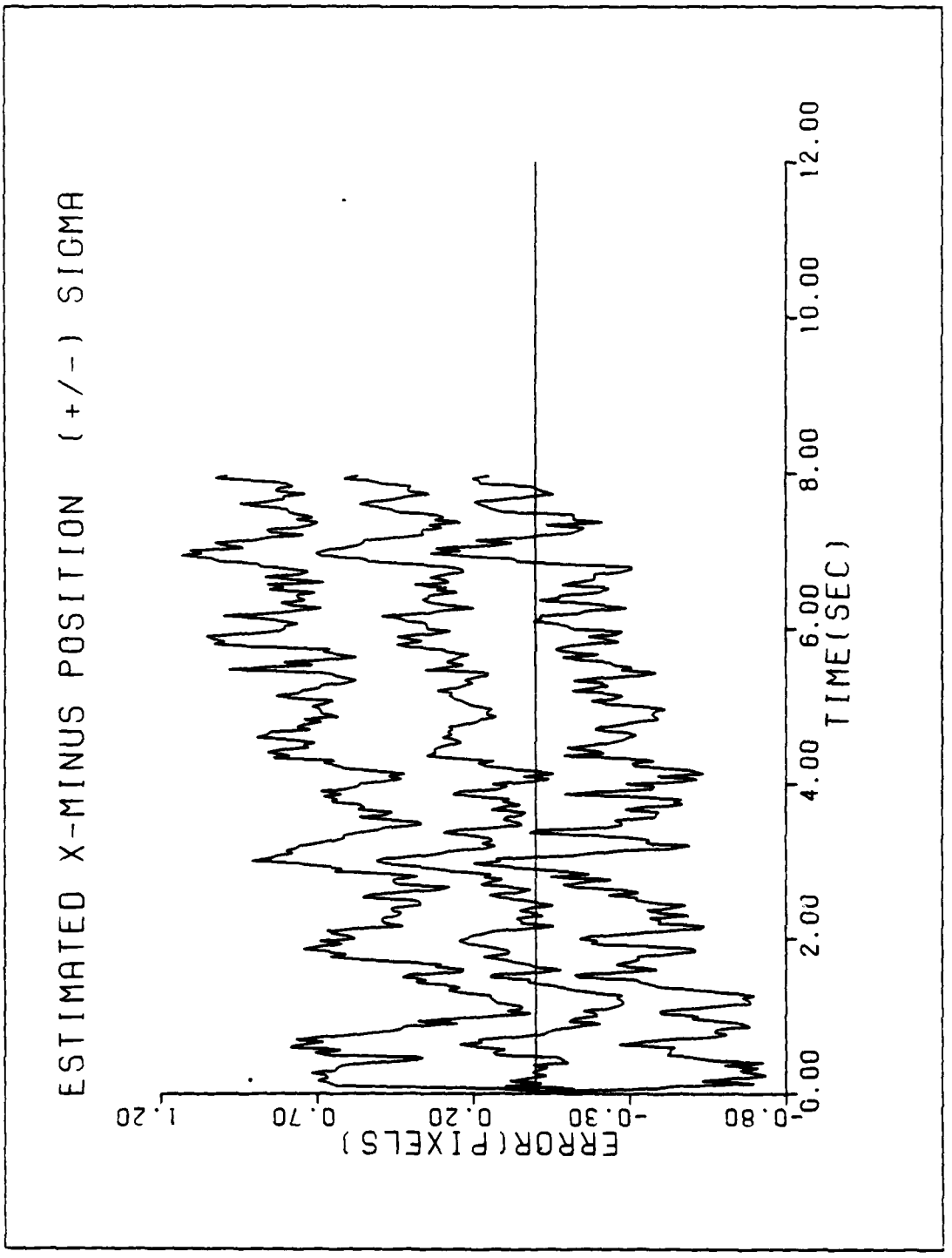


Figure D-1c. Performance Plot for TIMI-XT

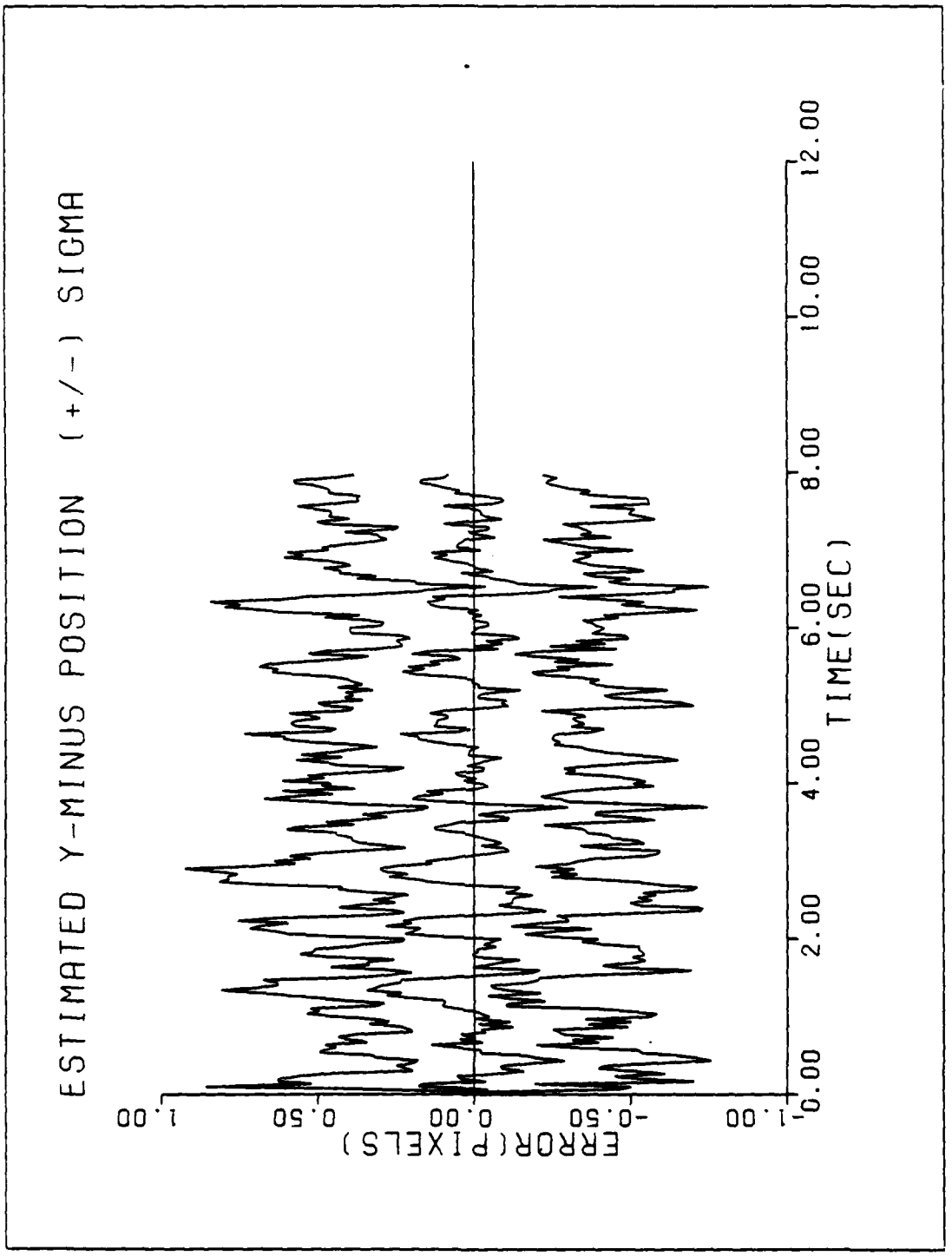


Figure D-1d. Performance Plot for TIM-IT

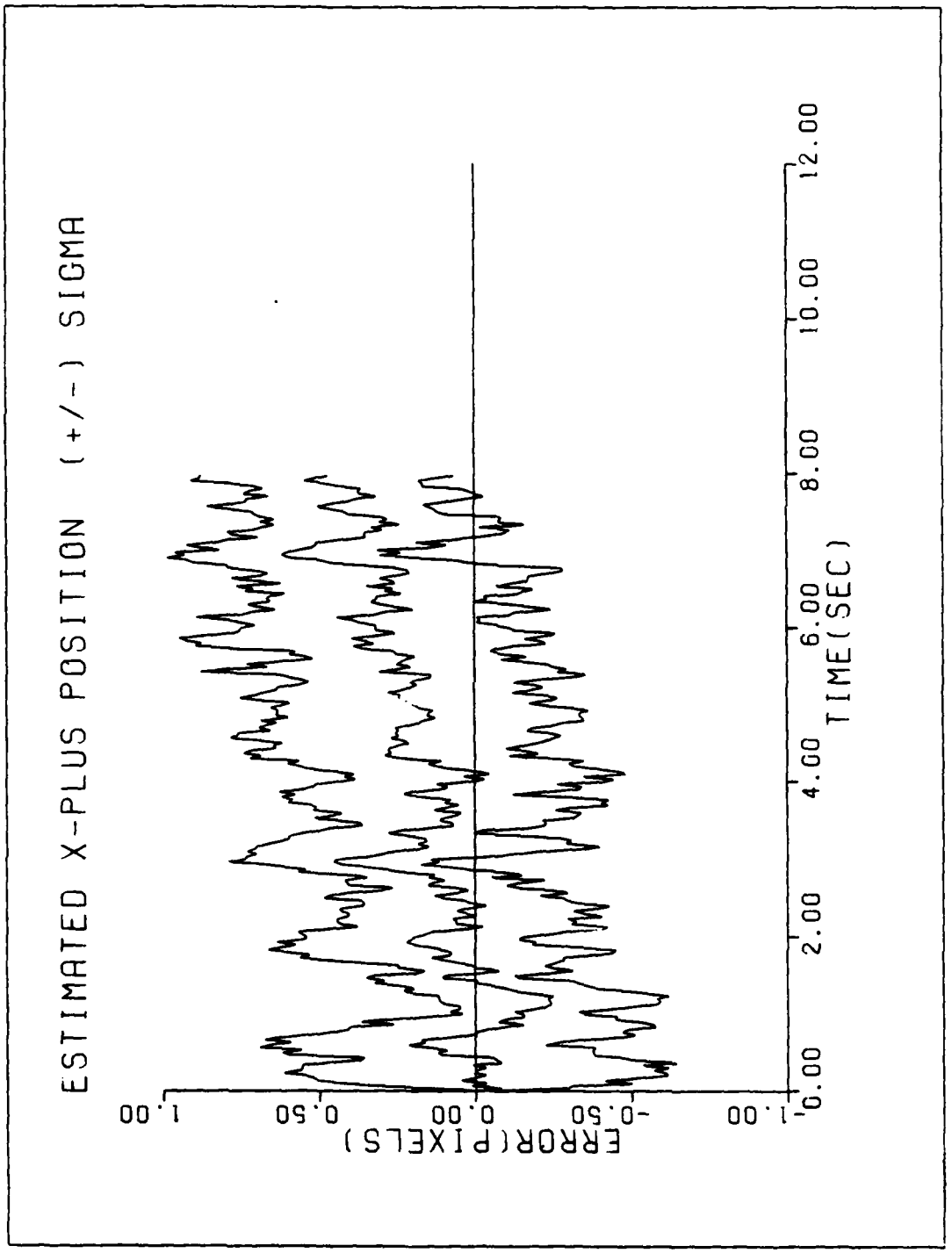


Figure D-1e. Performance Plot for TIMM-XT

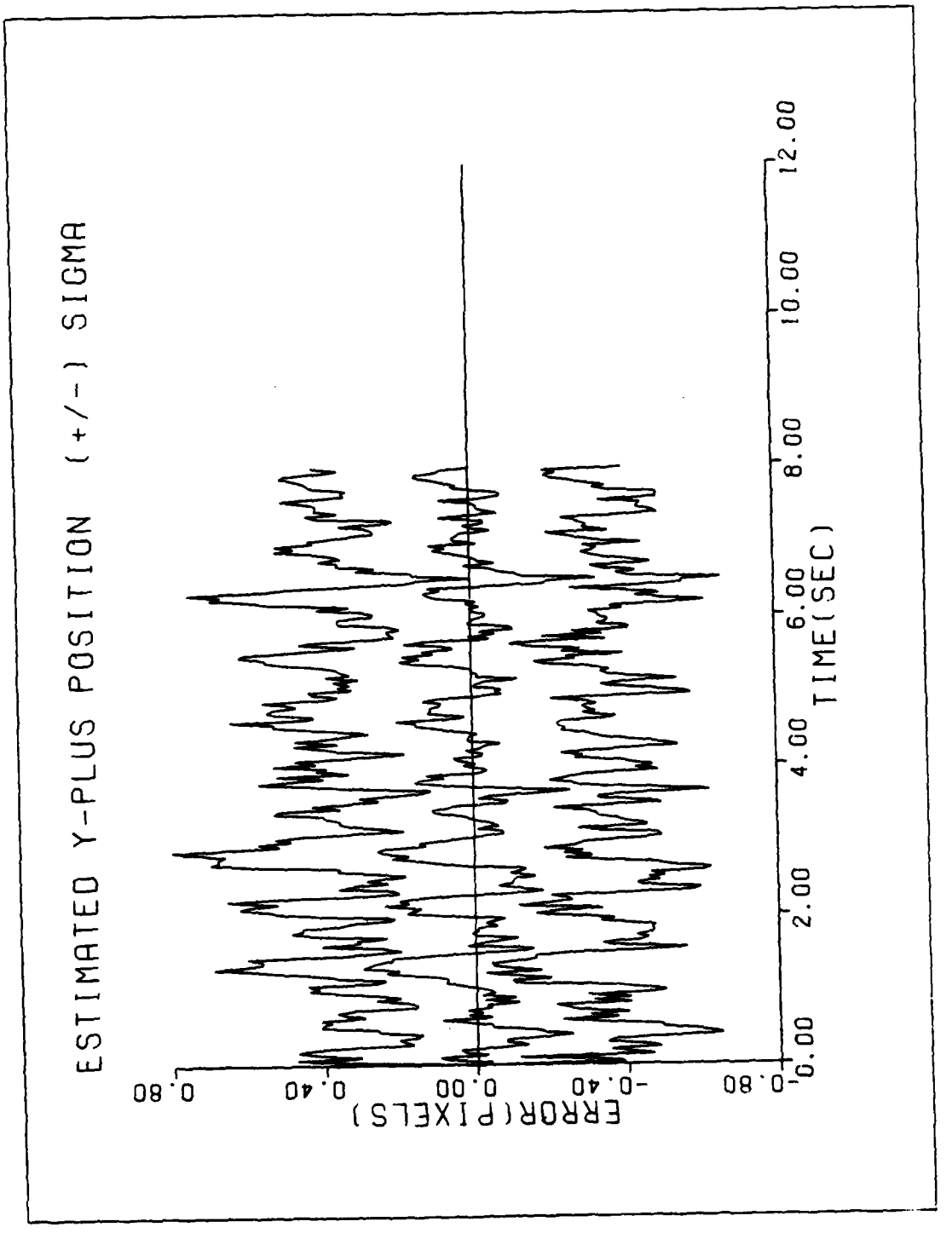


Figure D-1f. Performance Plot for TIMM-XT

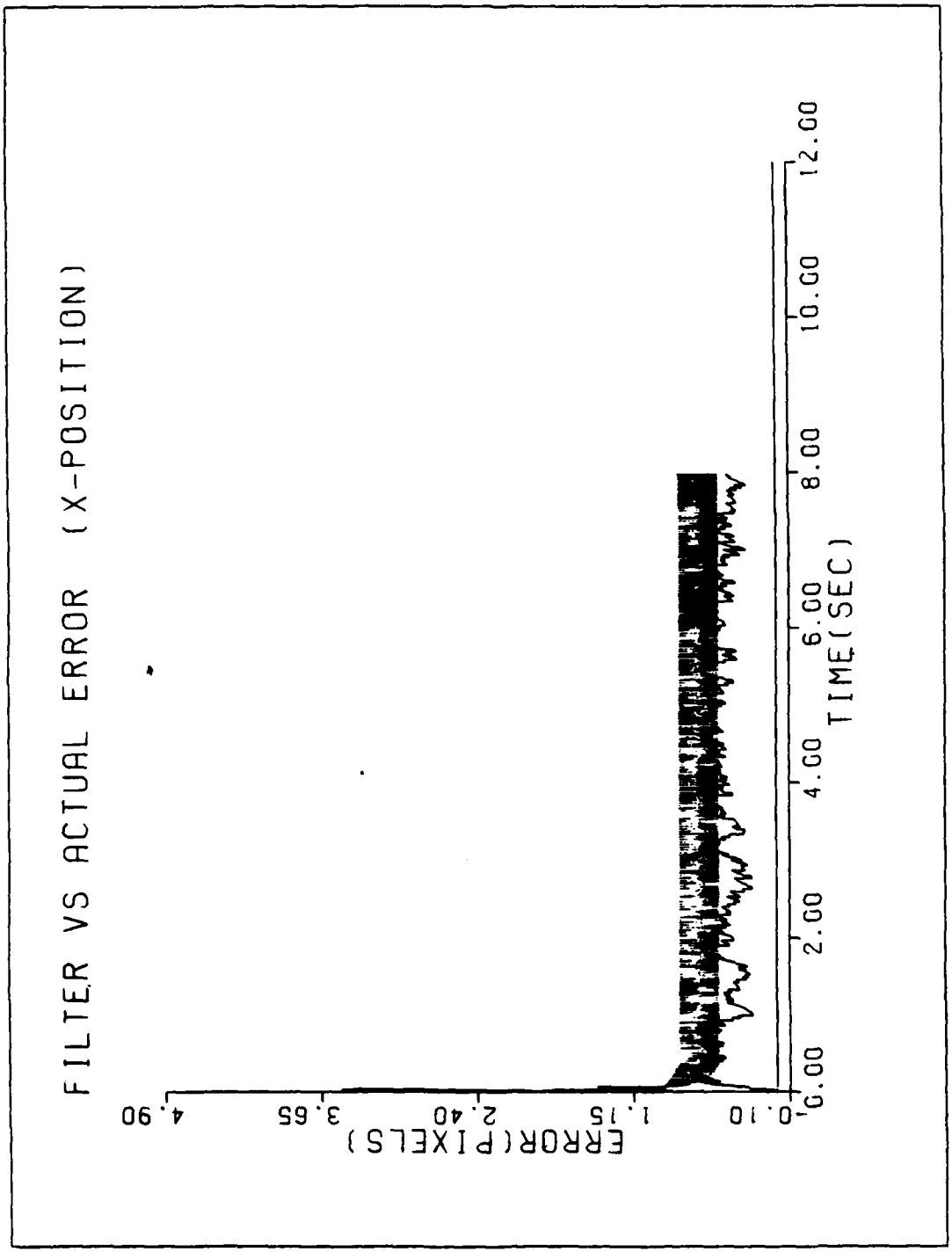


Figure D-2a. Performance Plot for T2G2MM-XT

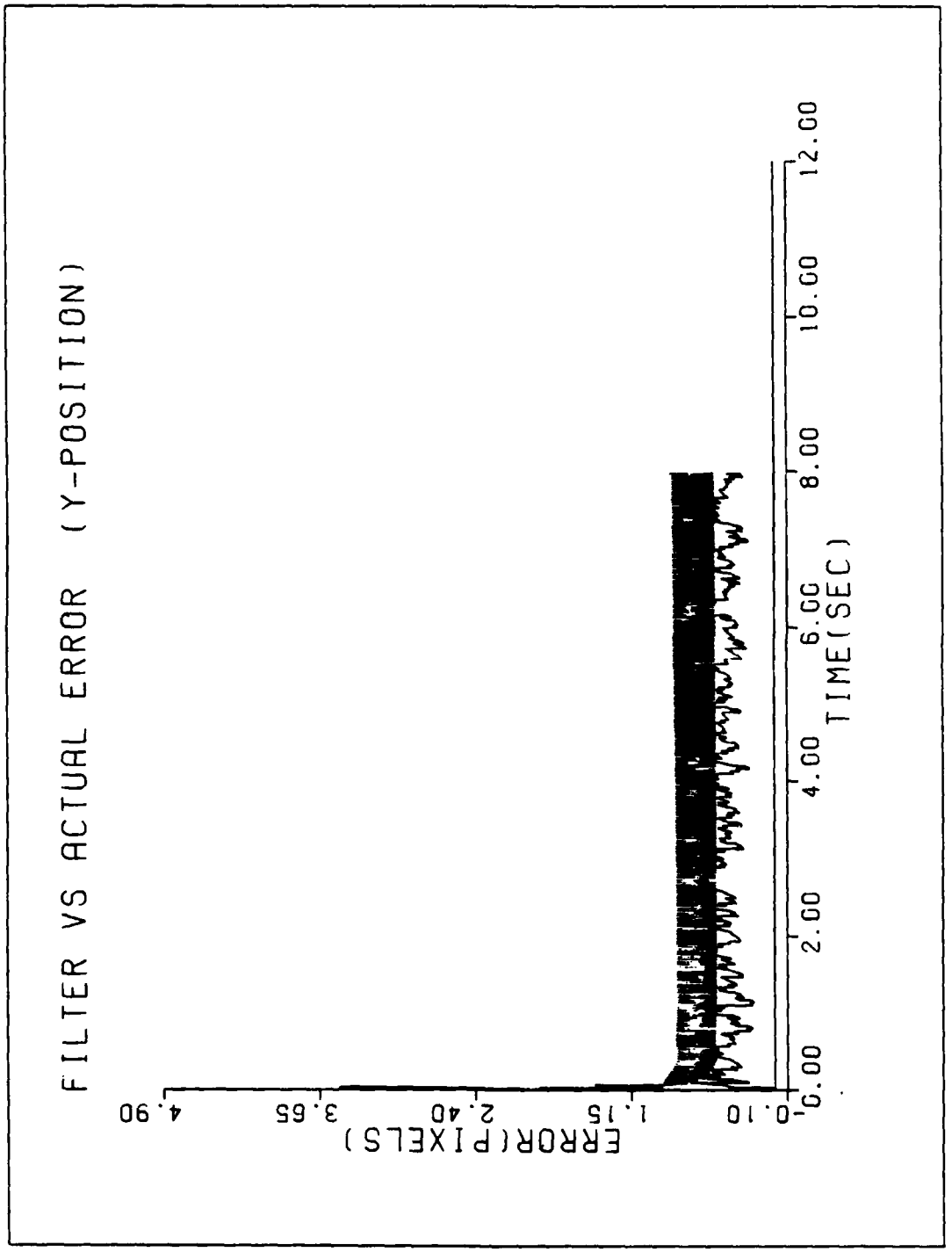


Figure D-2b. Performance Plot for T2G2MM-XT

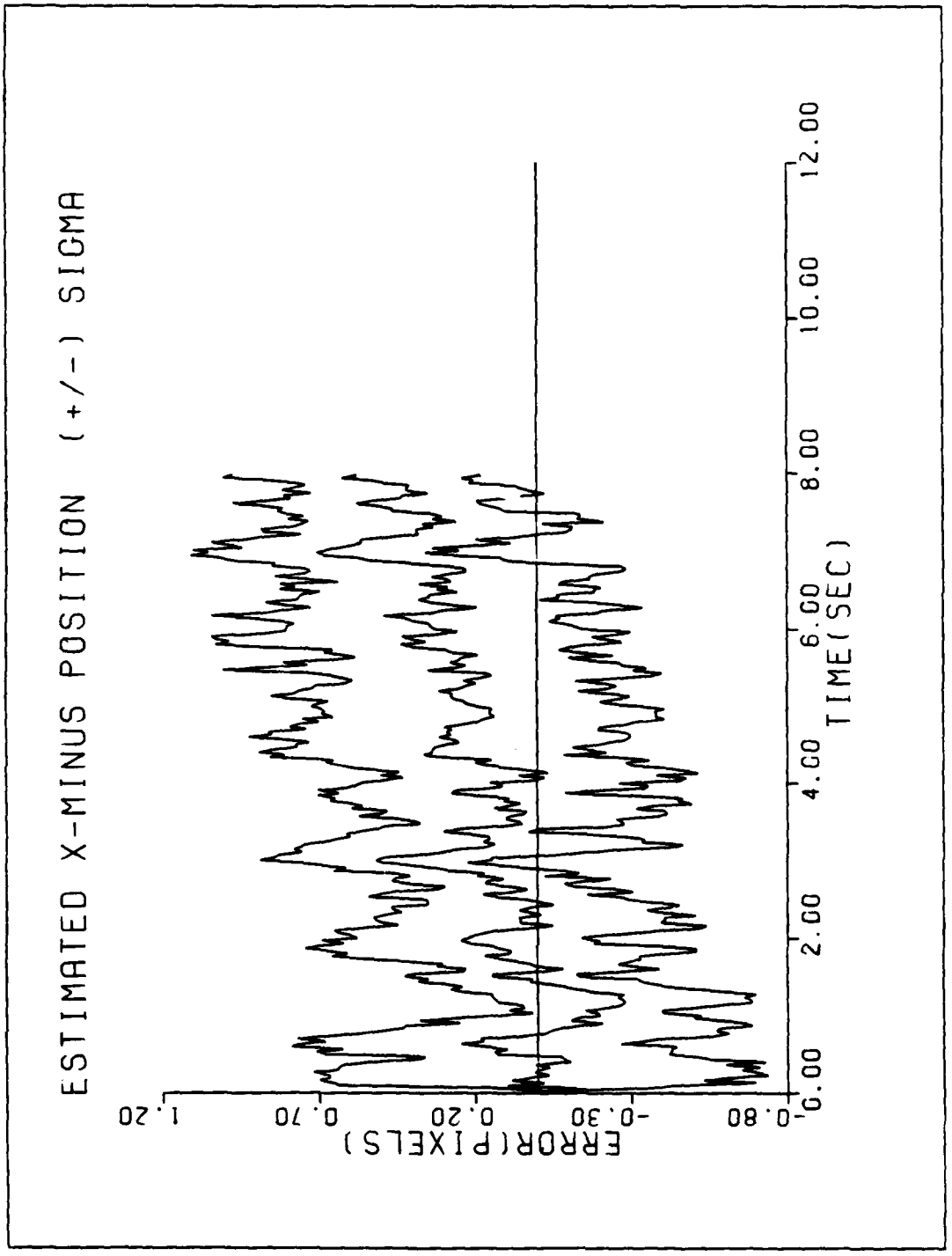


Figure D-2c. Performance Plot for T2G2MM-XT

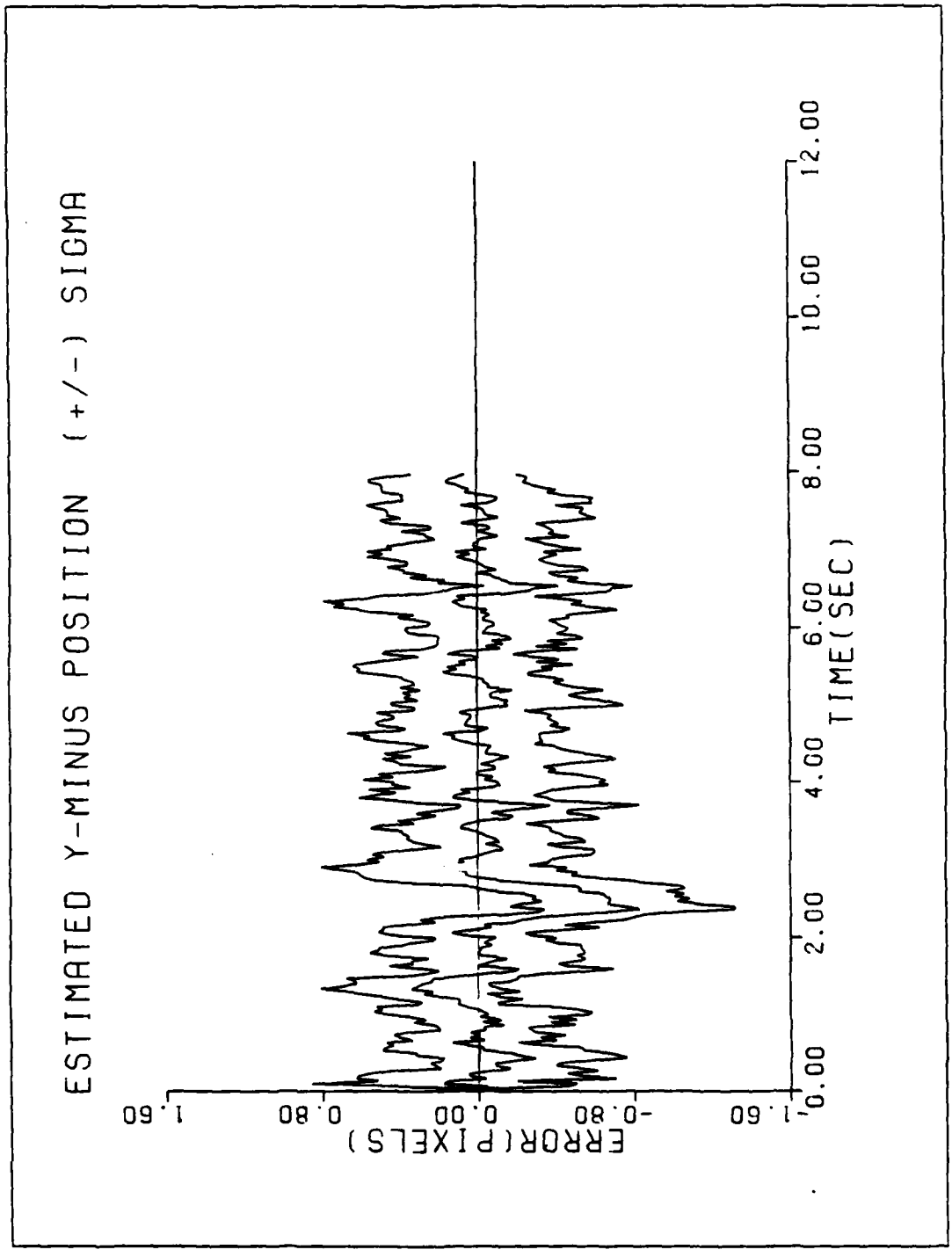


Figure D-2d. Performance Plot for T2G2MM-XT

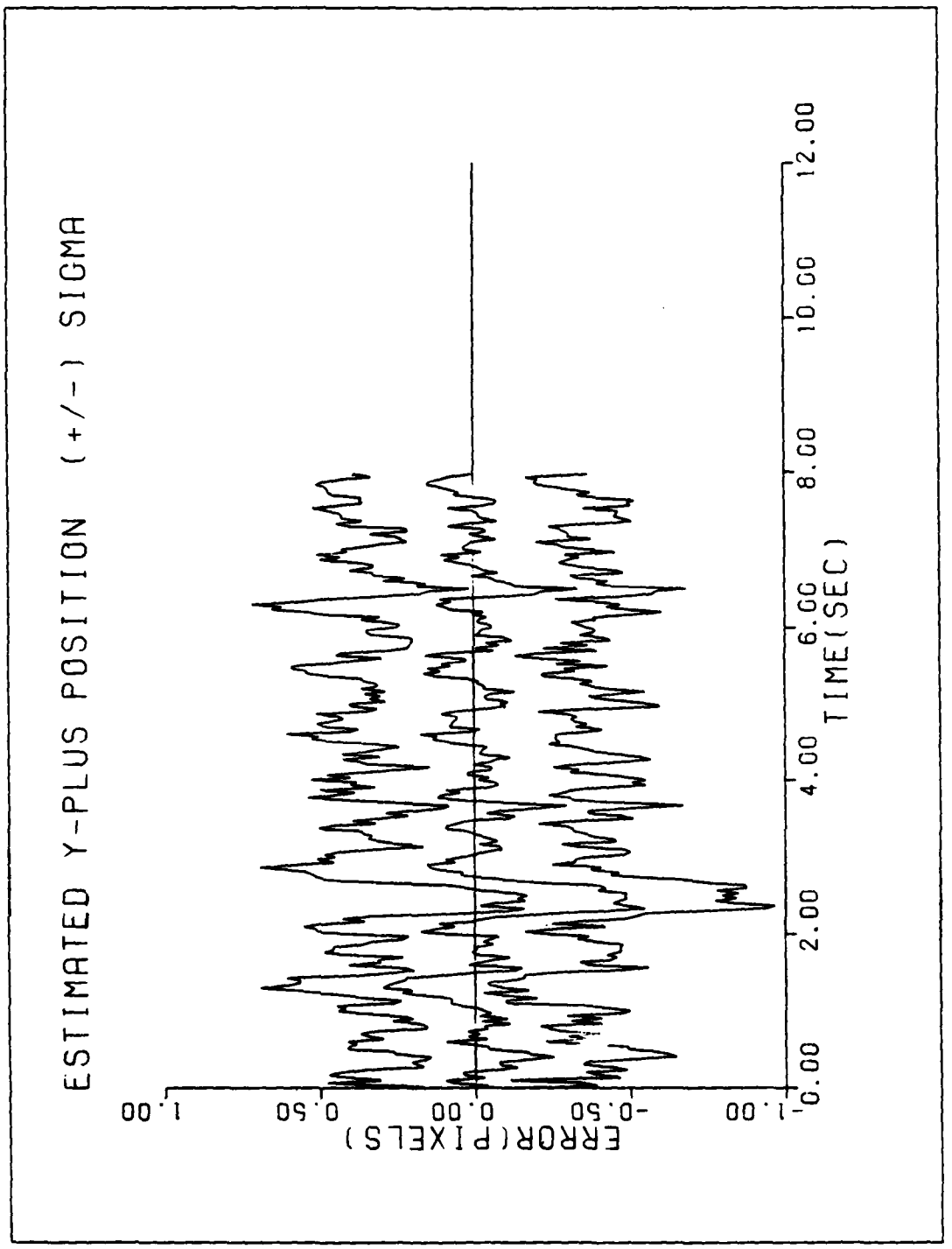


Figure D-2e. Performance Plot for T2G2MM-XT

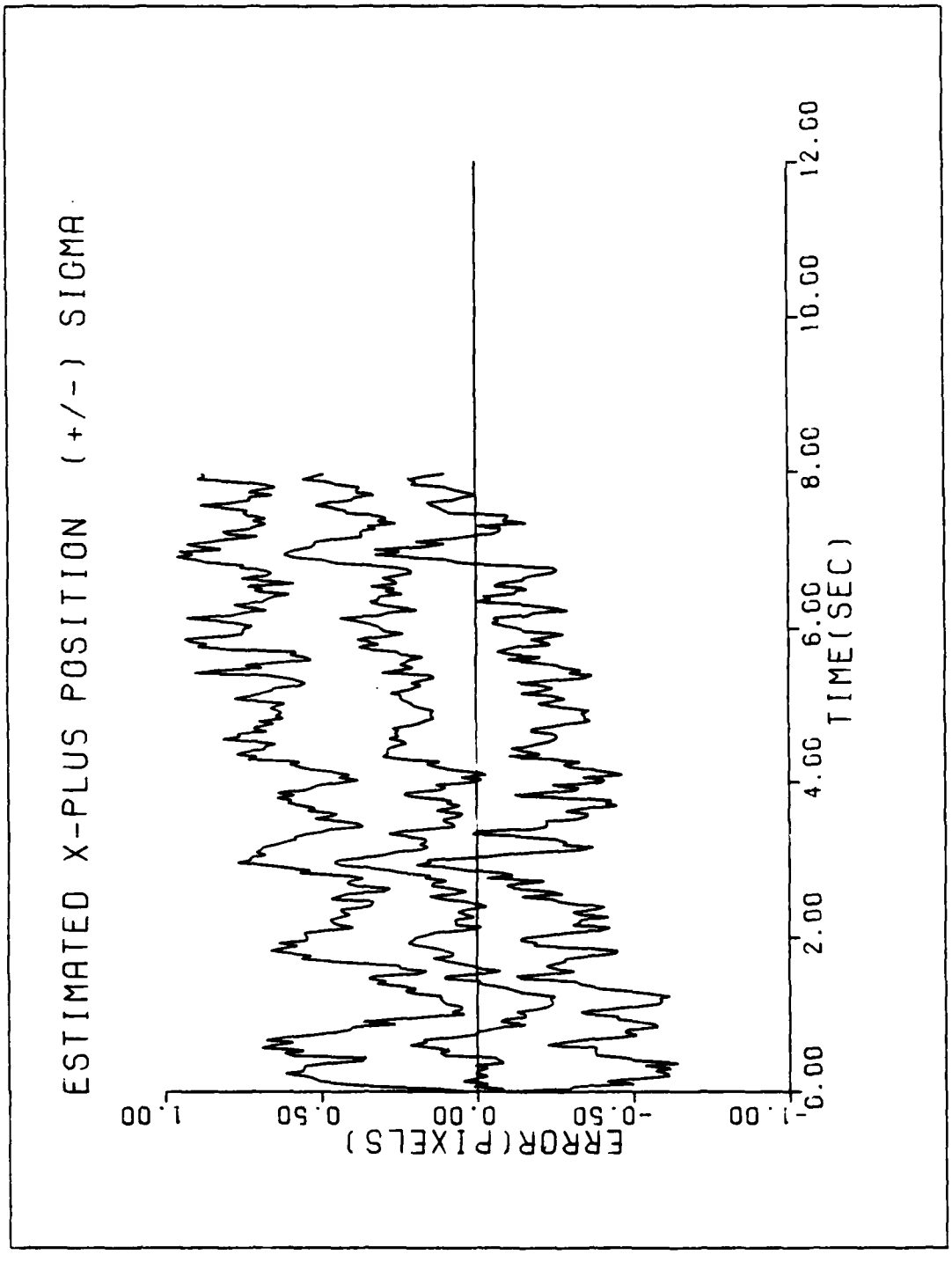


Figure D-2f. Performance Plot for T2G2MM-XT

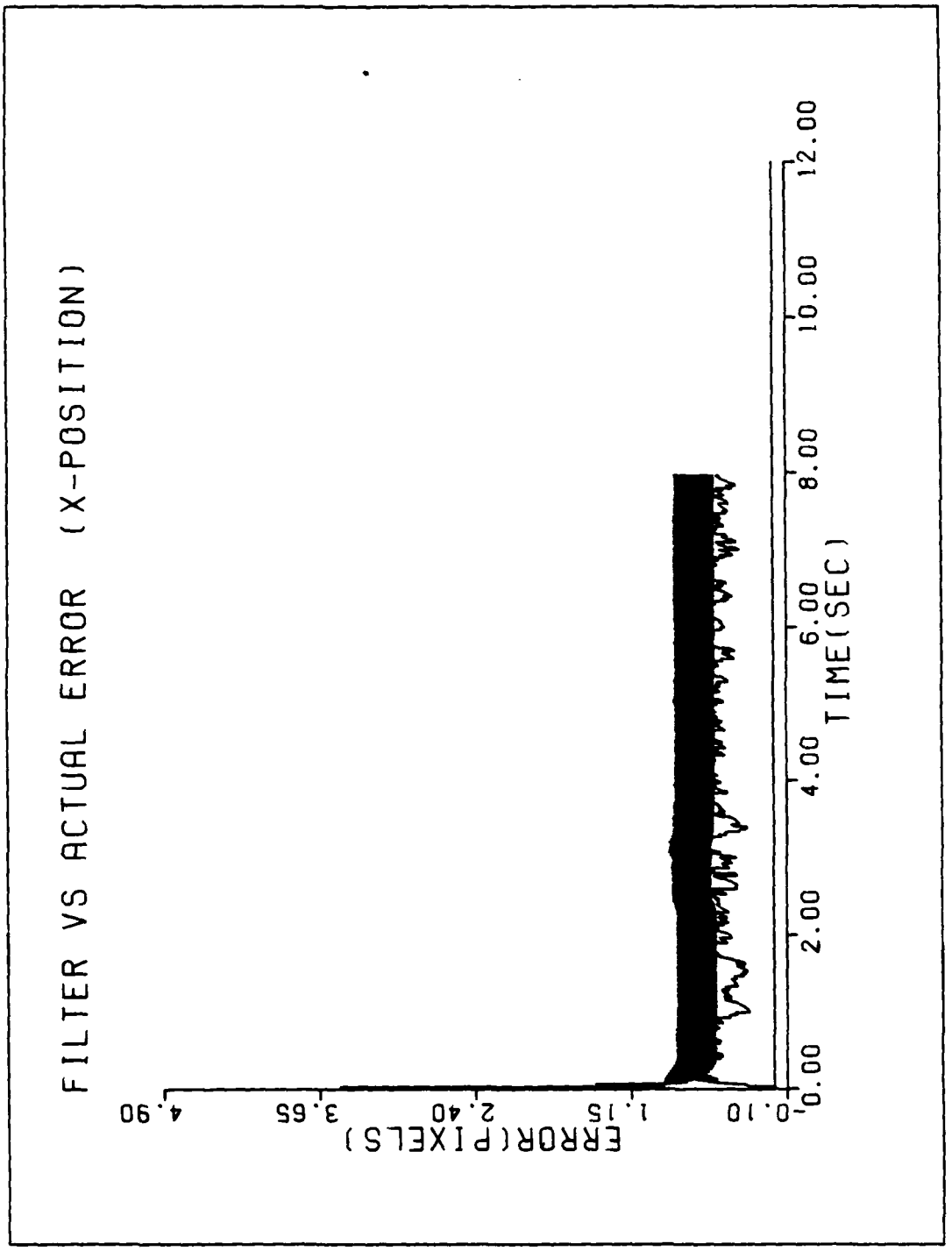


Figure D-3a. Performance Plot for T2G10MM-XT

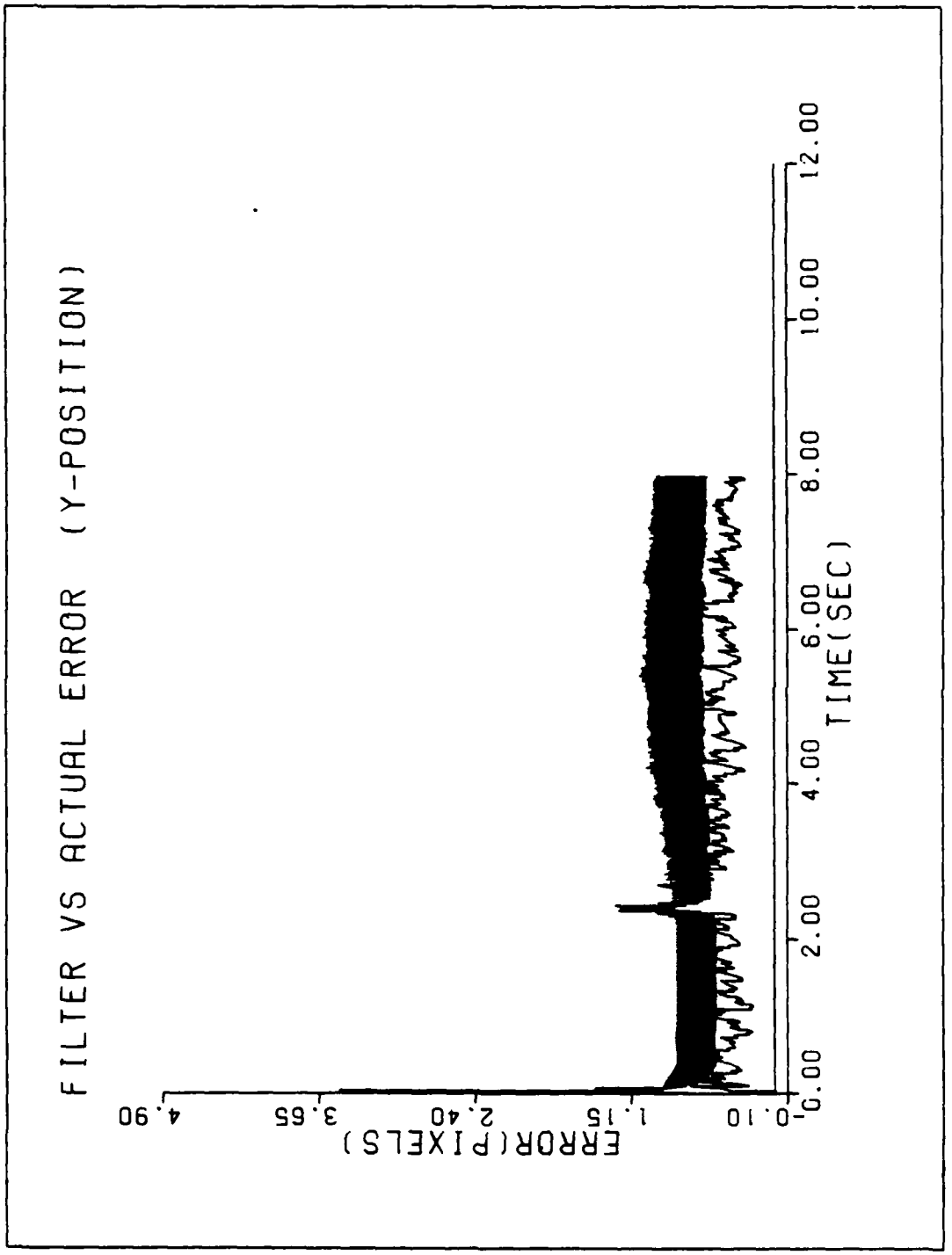


Figure D-3b. Performance Plot for T2G10MM-XT

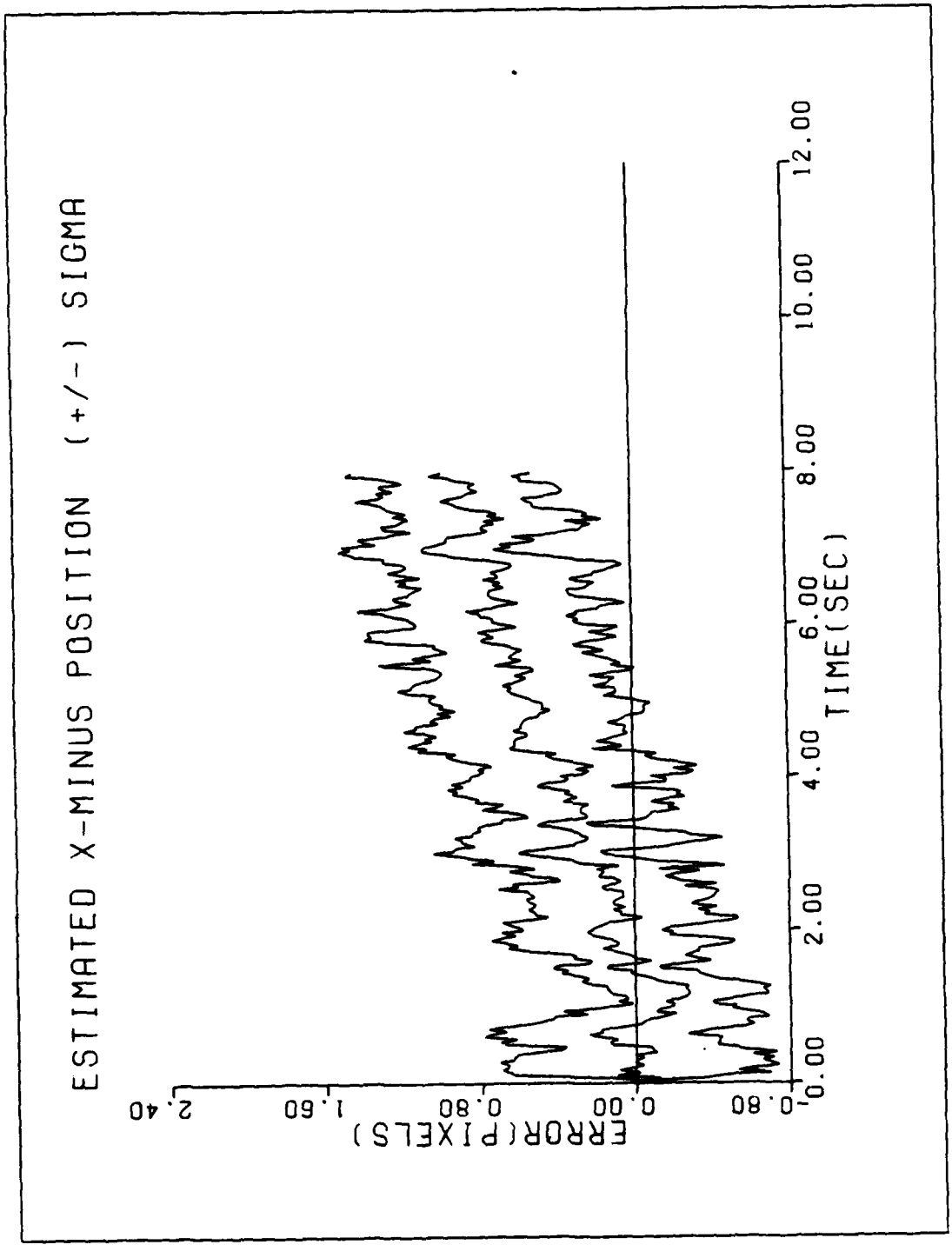


Figure D-3c. Performance Plot for T2G10MM-XT

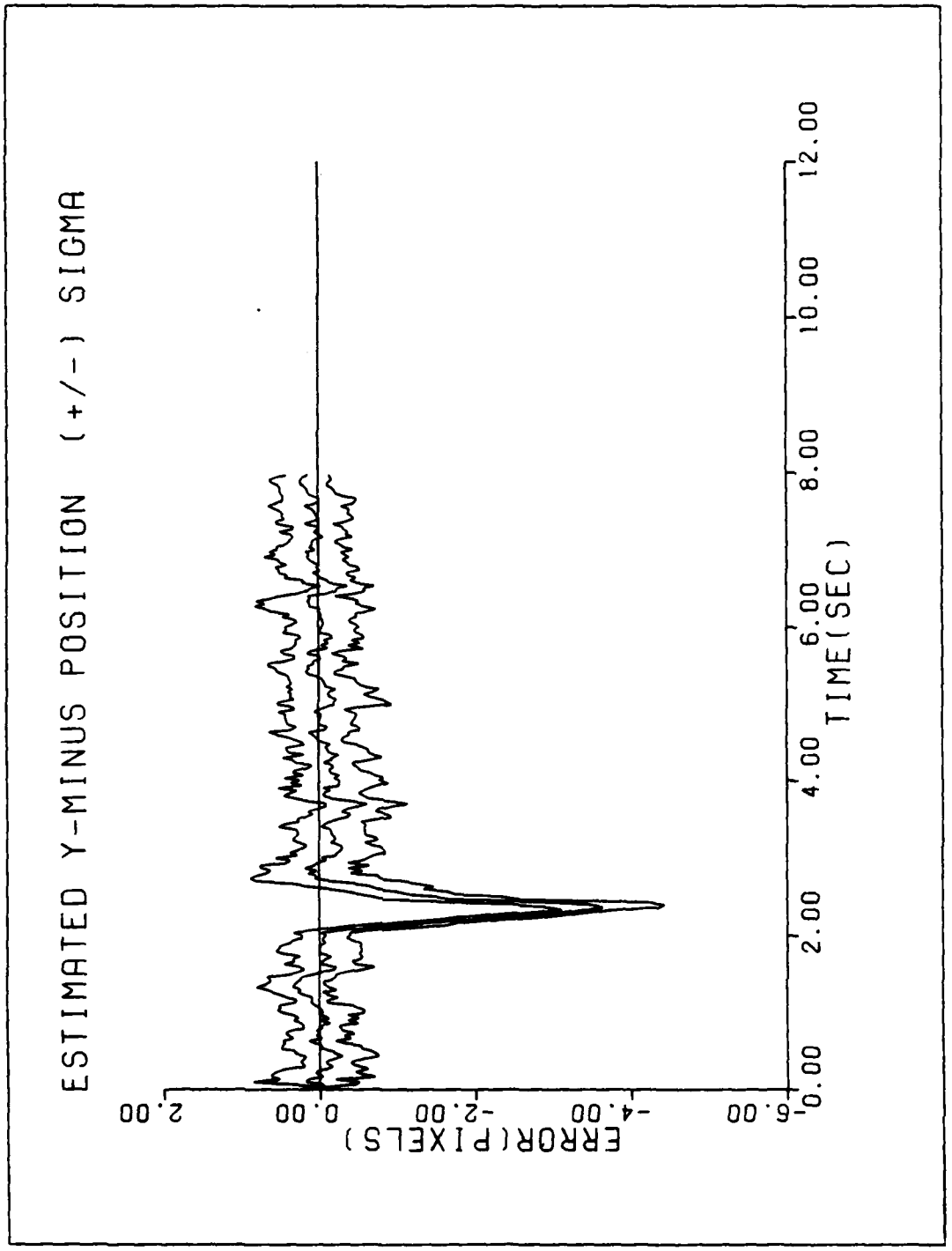


Figure D-3d. Performance Plot for T2G10MM-XT

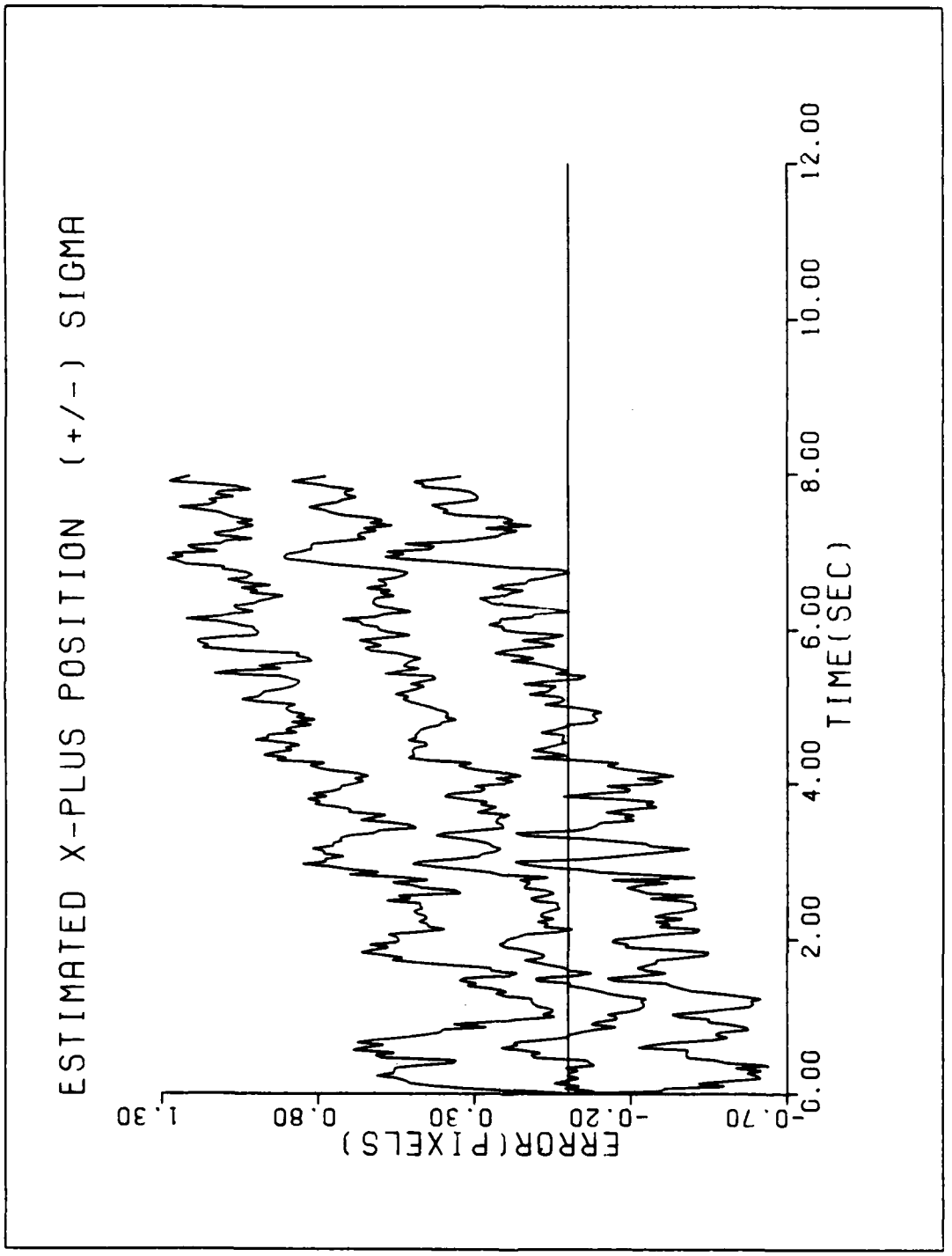


Figure D-3e. Performance Plot for T2G10MM-XT

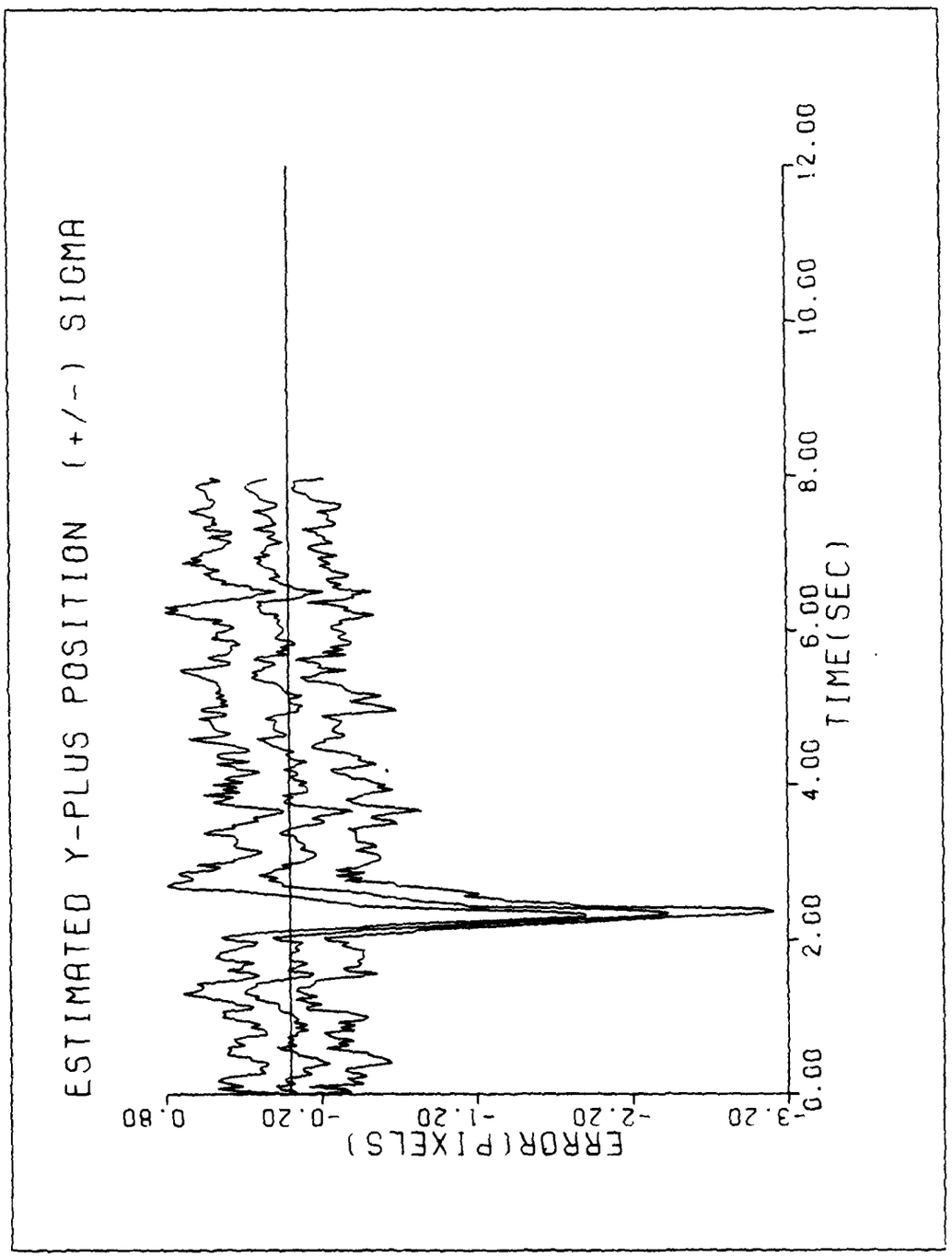


Figure D-3f. Performance Plot for T2G10MM-XT

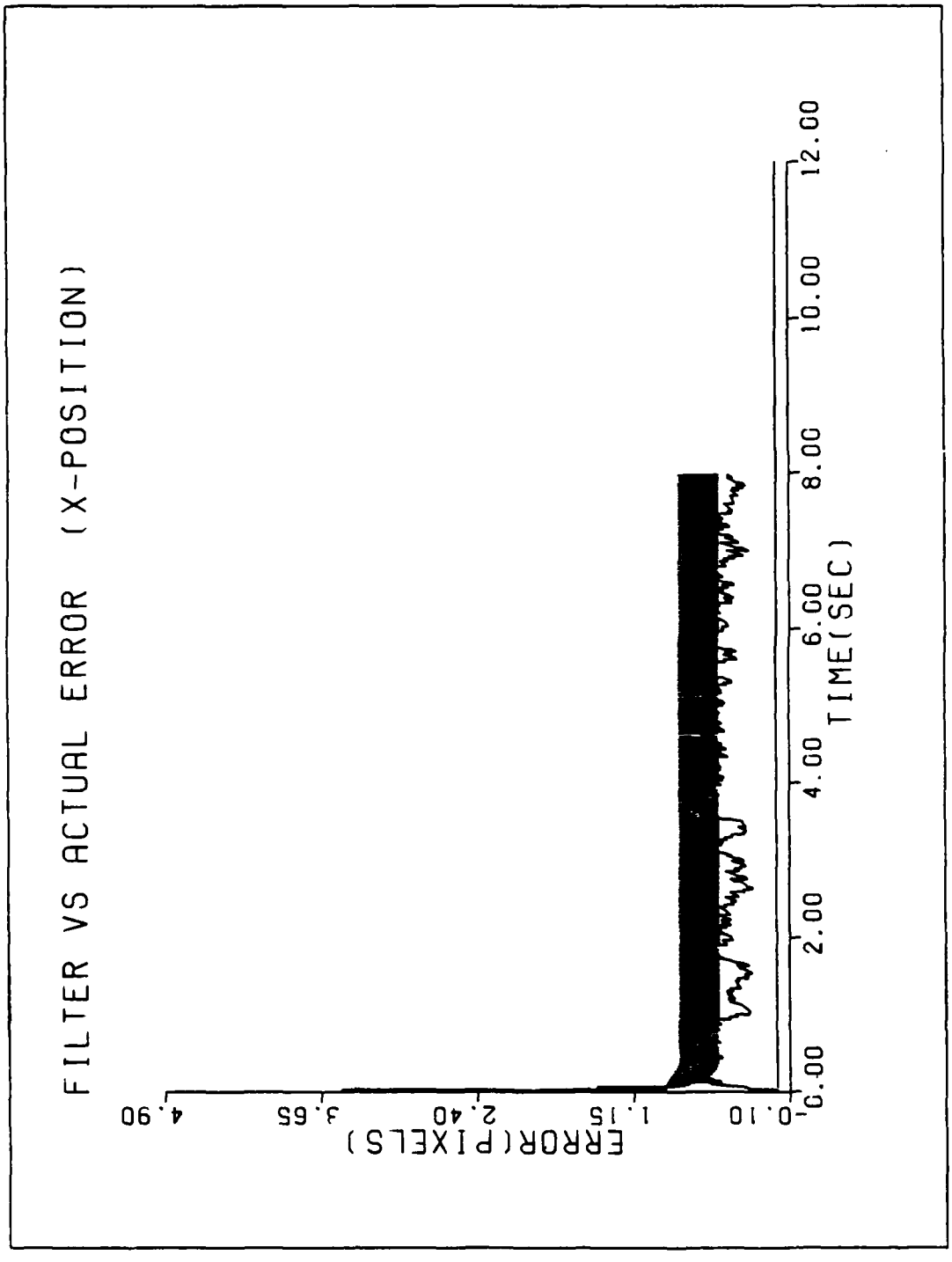


Figure D-4a. Performance Plot for T3G2MM-XT

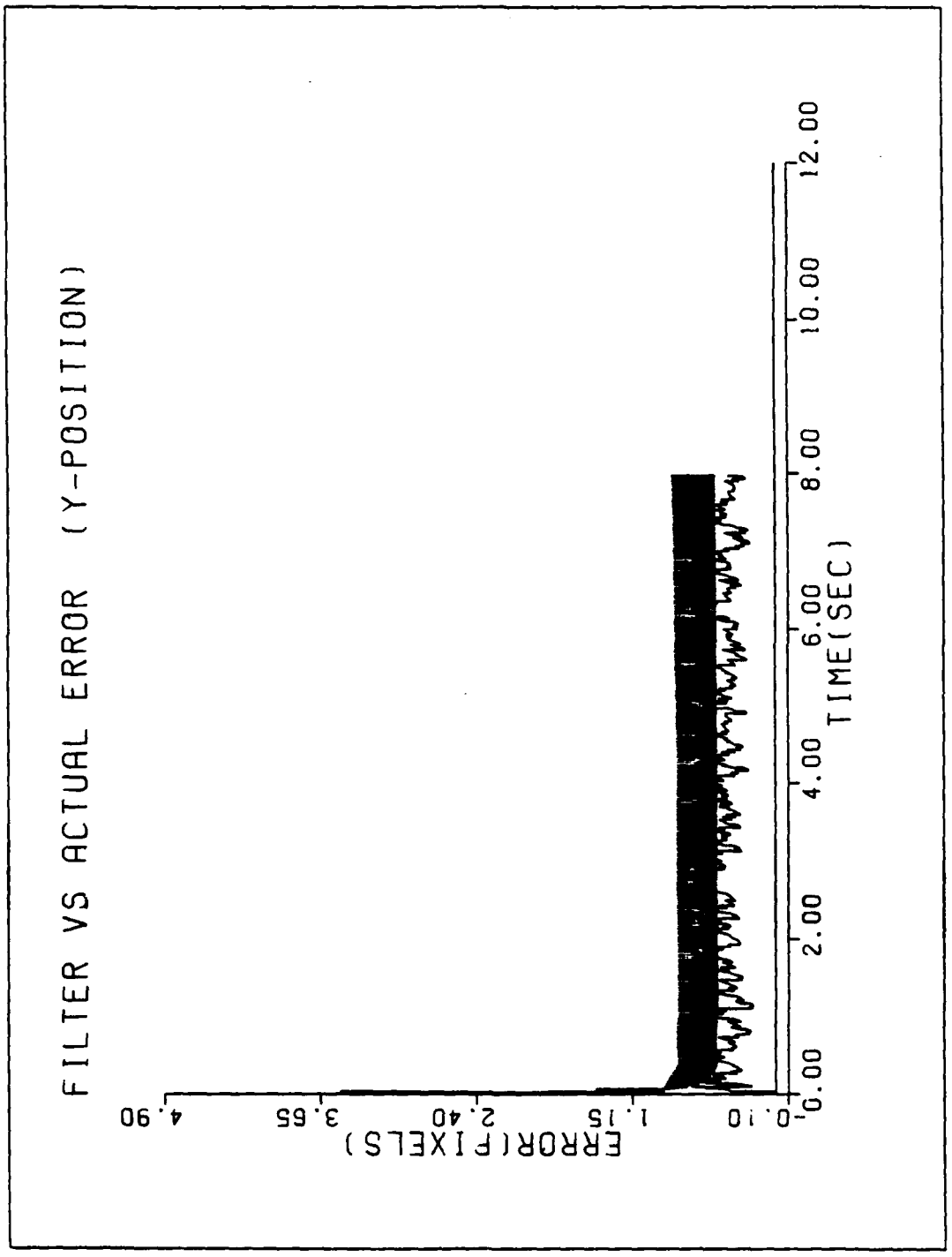


Figure D-4b. Performance Plot for T3G2MM-XT

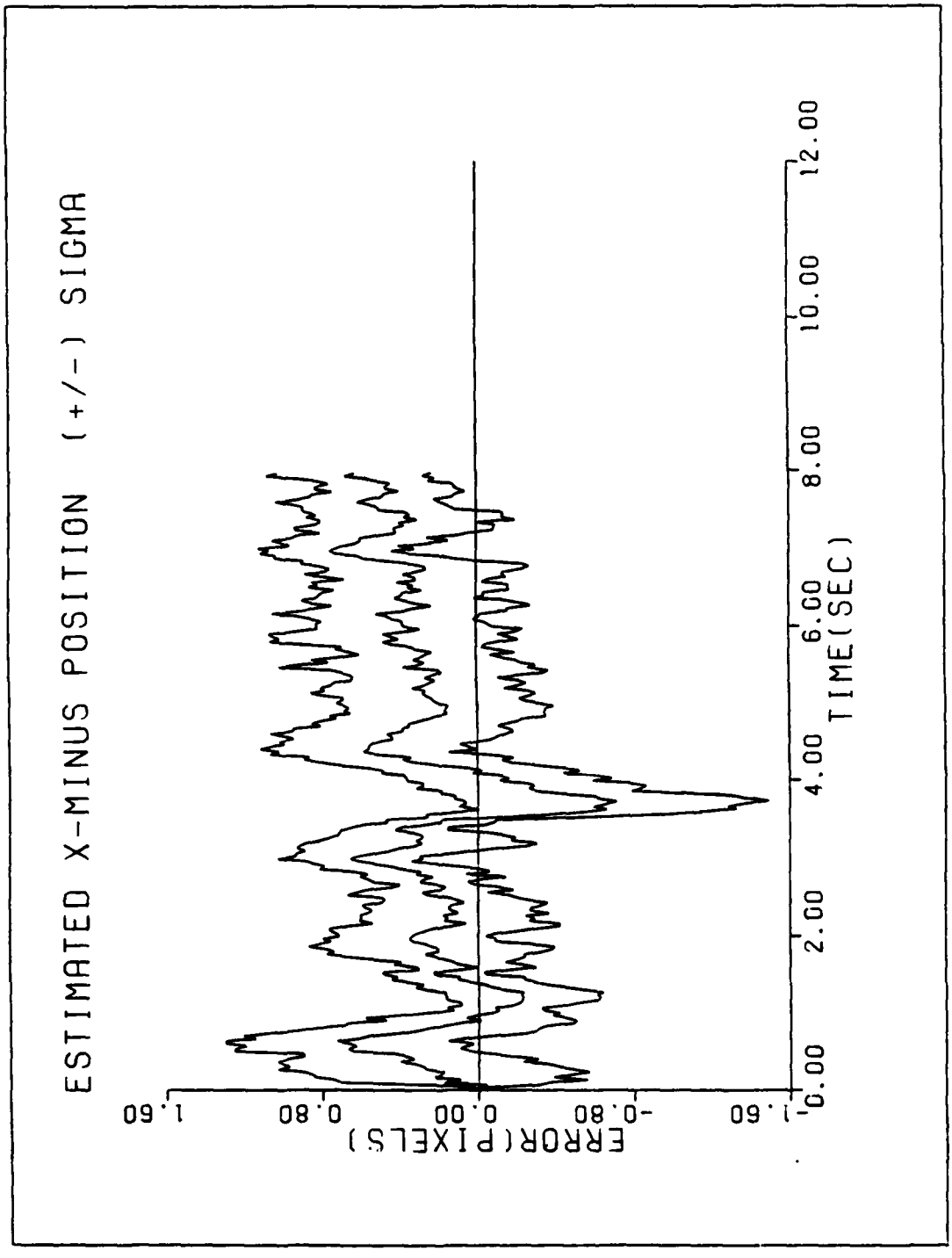


Figure D-4c. Performance Plot for T3G2MM-XT

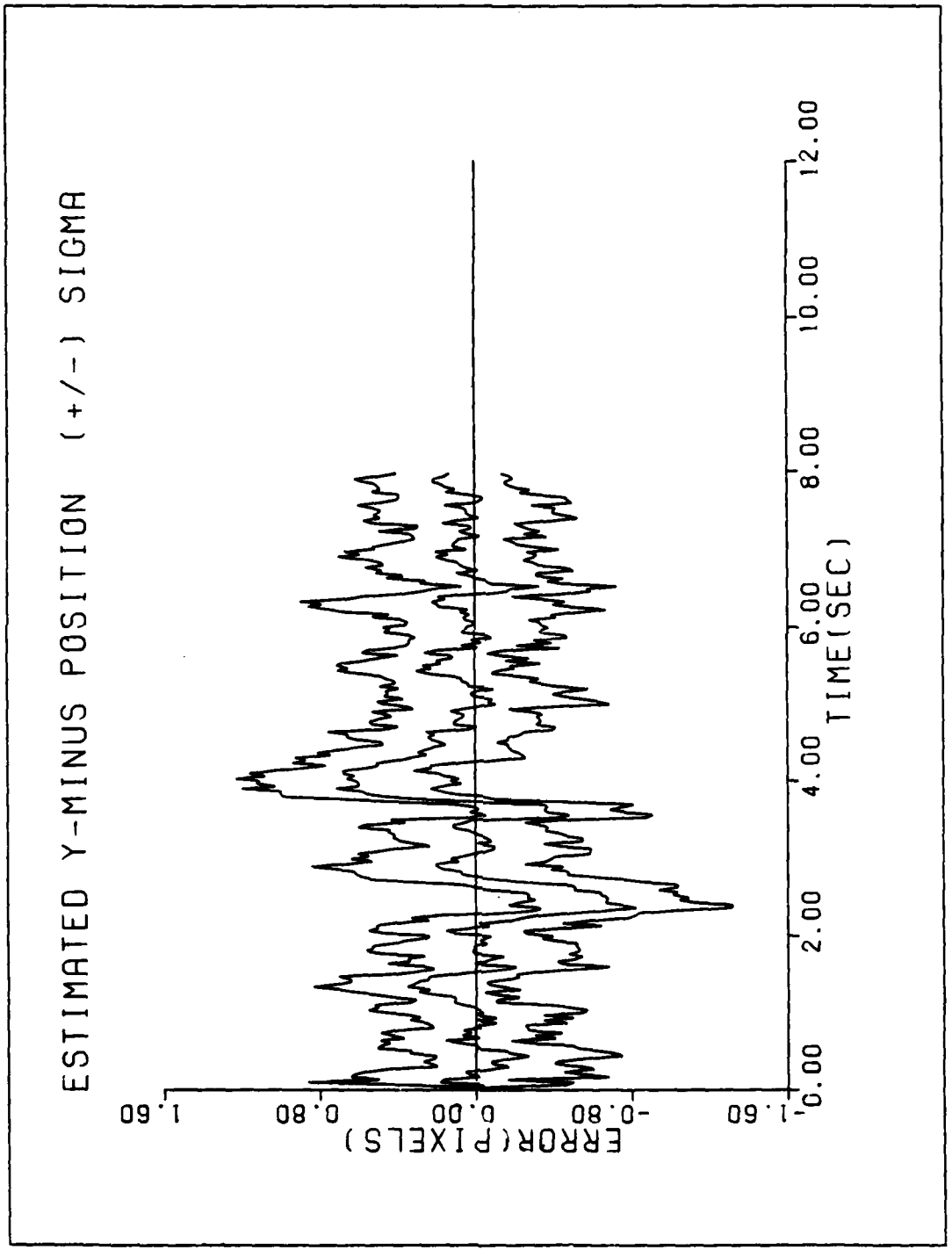


Figure D-4d. Performance Plot for T3G2MM-XT

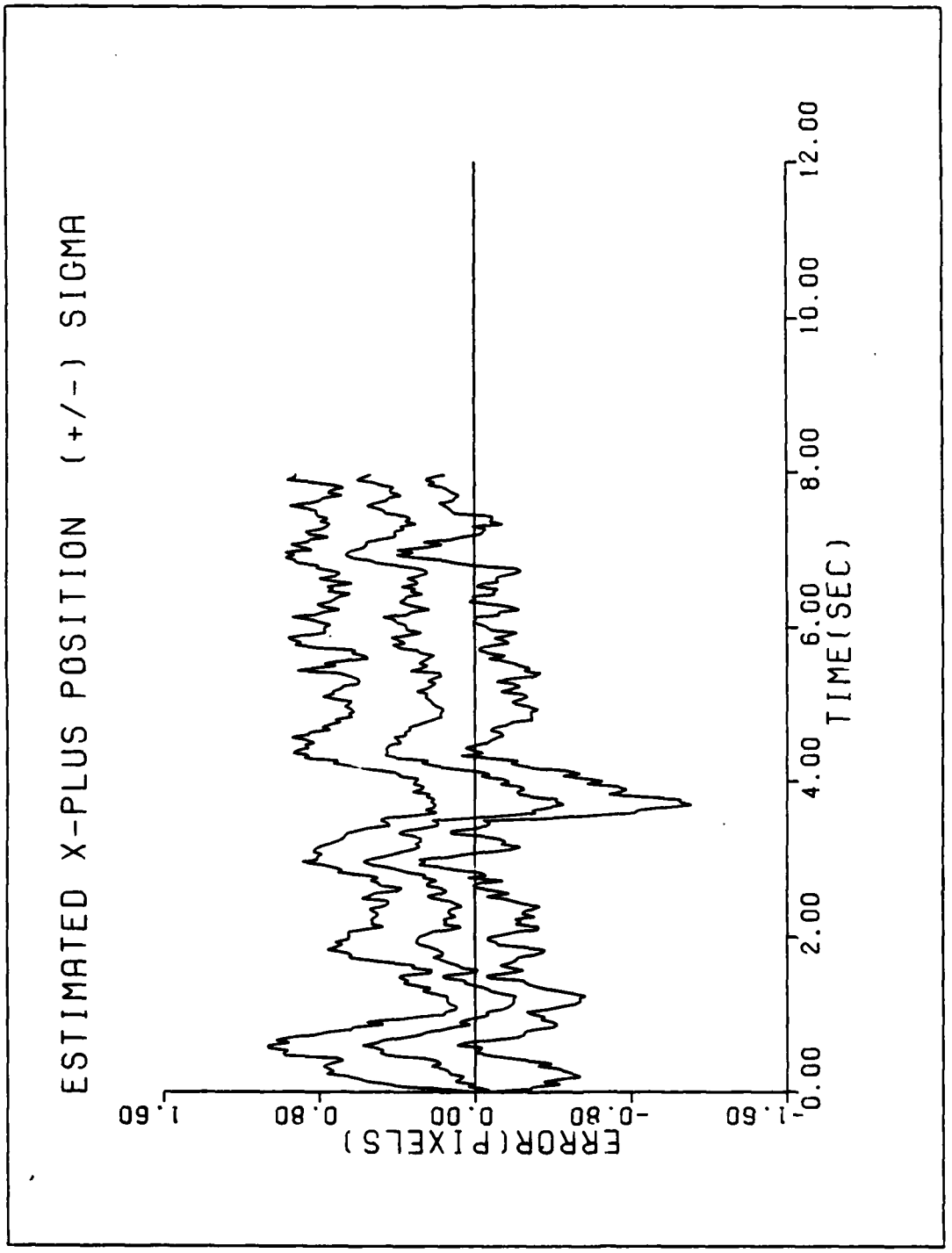


Figure D-4e. Performance Plot for T3G2MM-XT

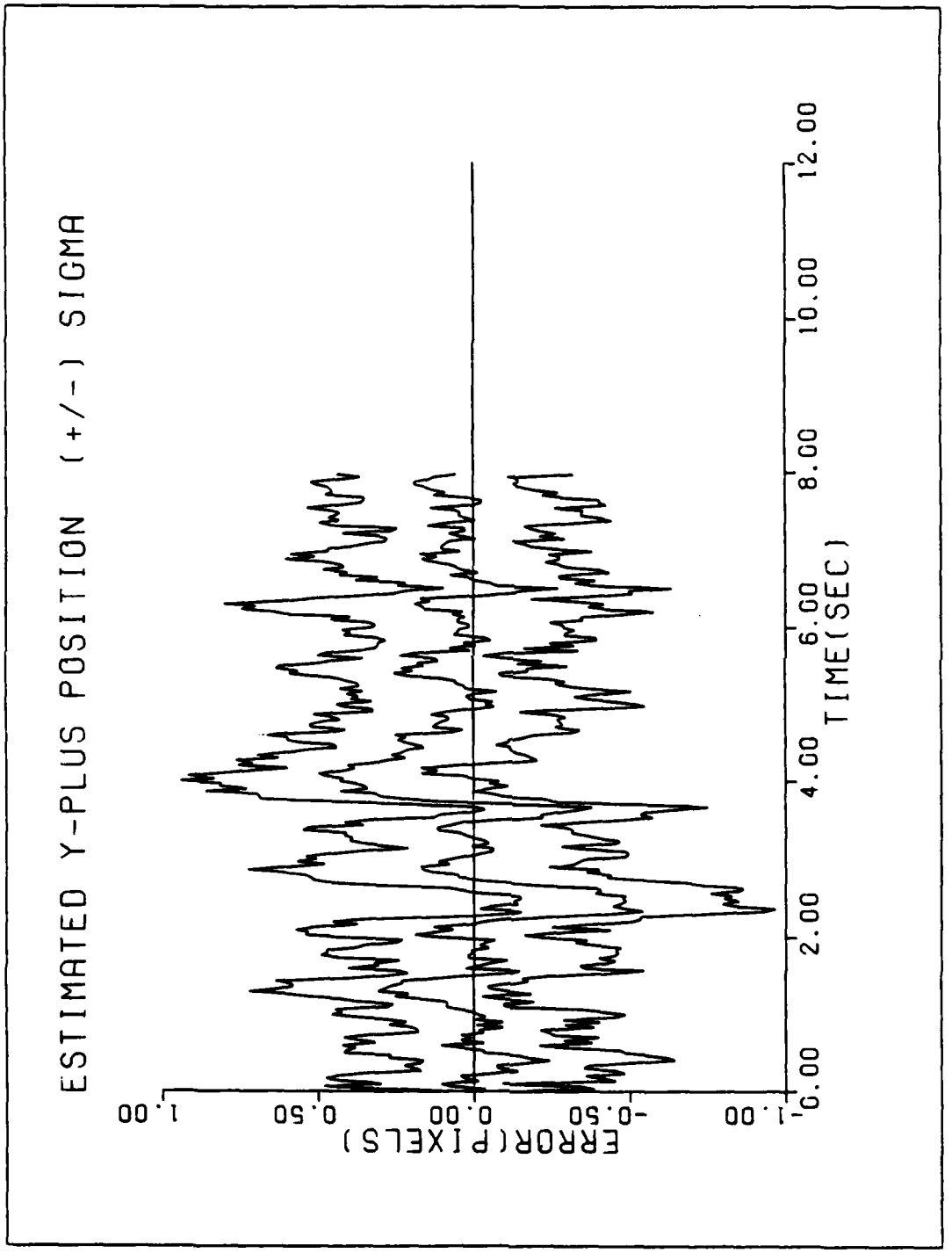


Figure D-4f. Performance Plot for T3G2MM-XT

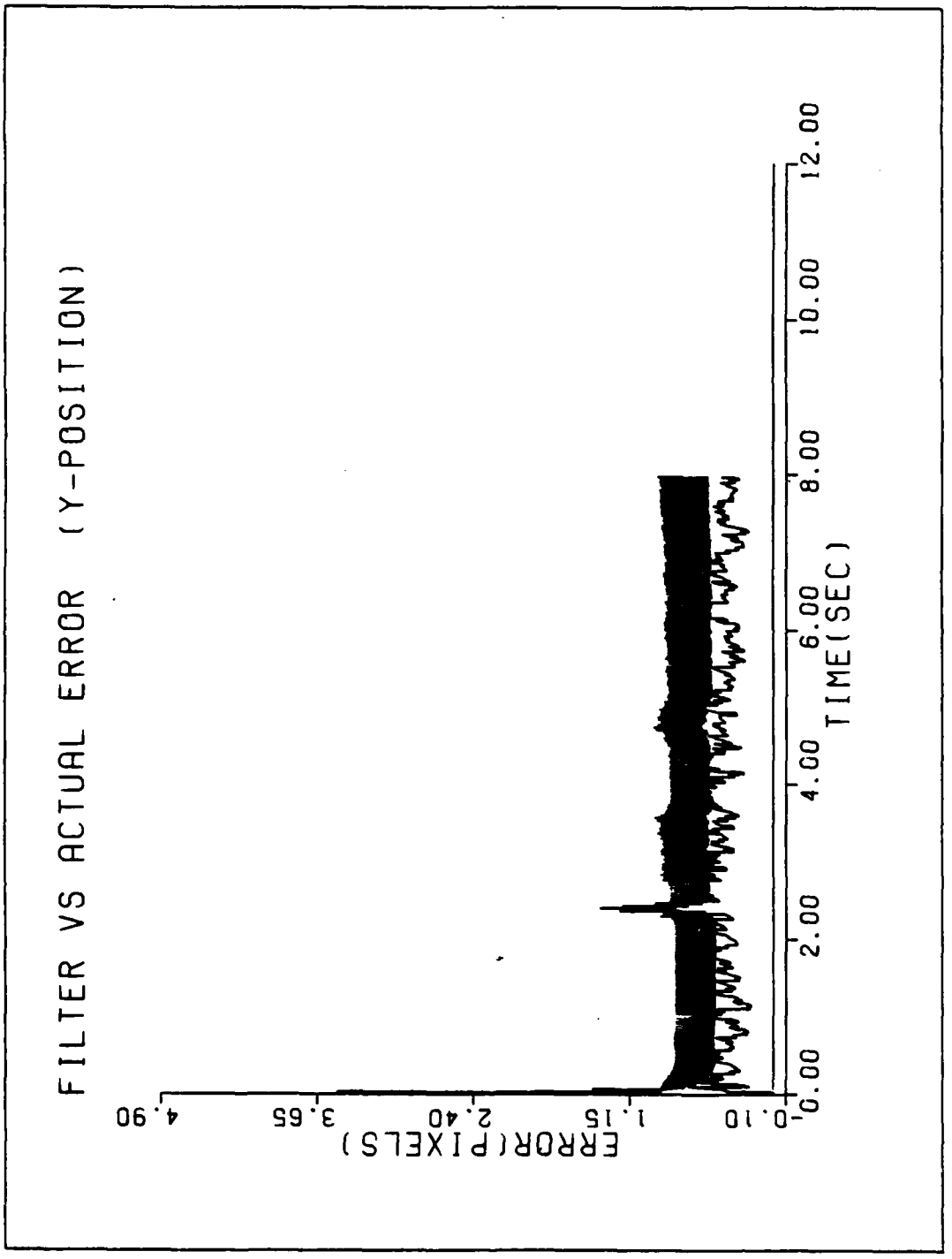


Figure D-5a. Performance Plot for T3G10MM-XT

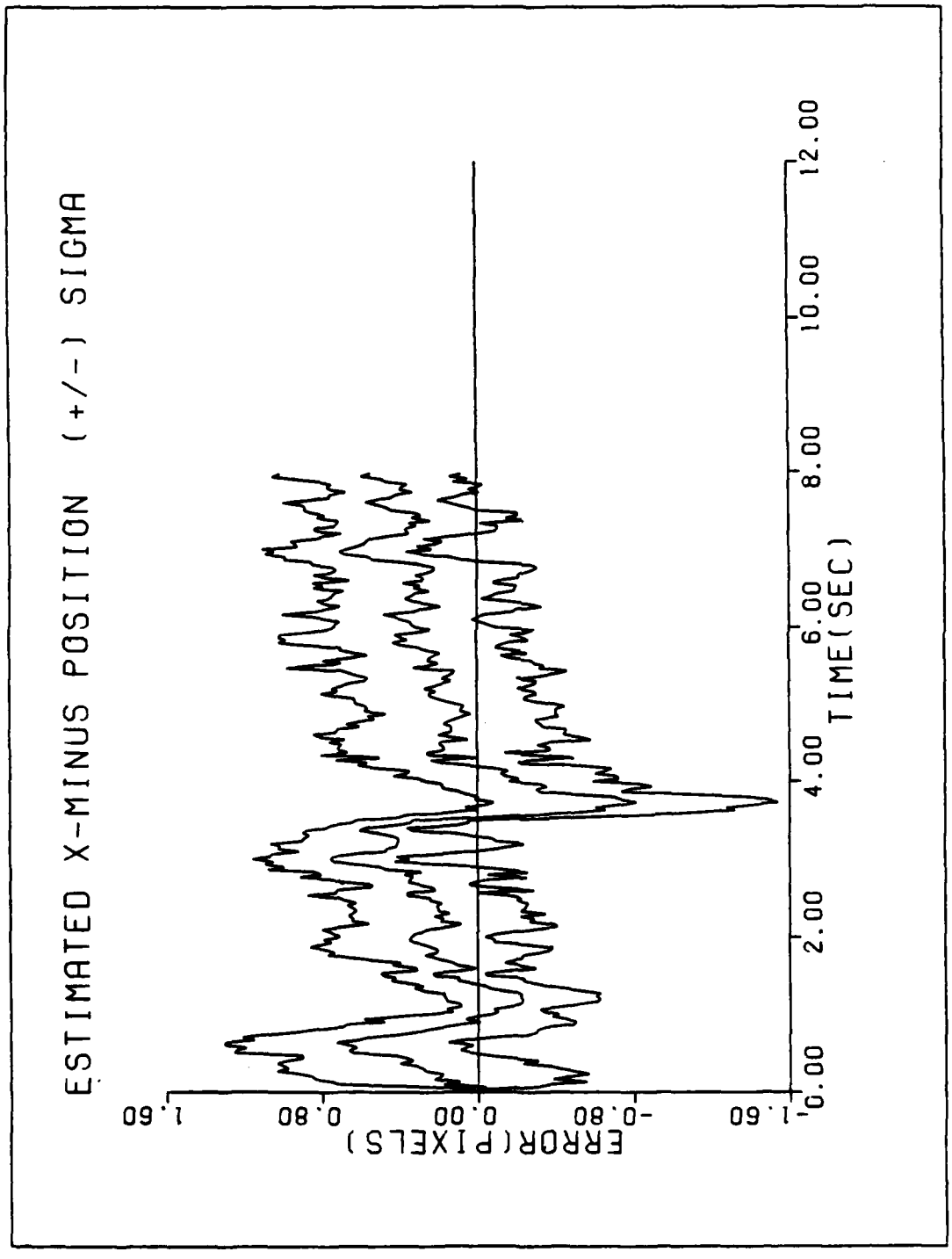


Figure D-5b. Performance Plot for T3G10MM-XT

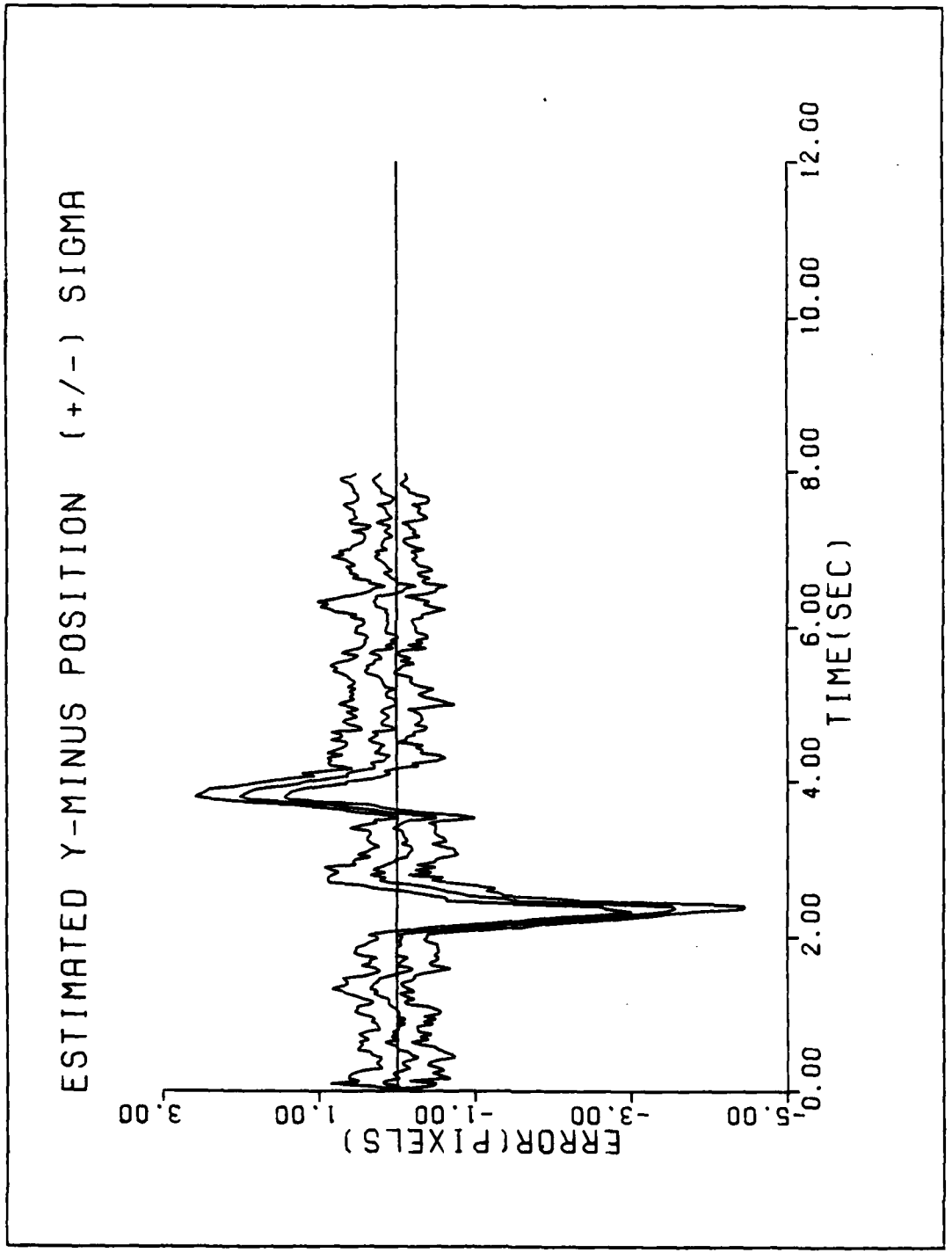


Figure D-5c. Performance Plot for T3G10MM-XT

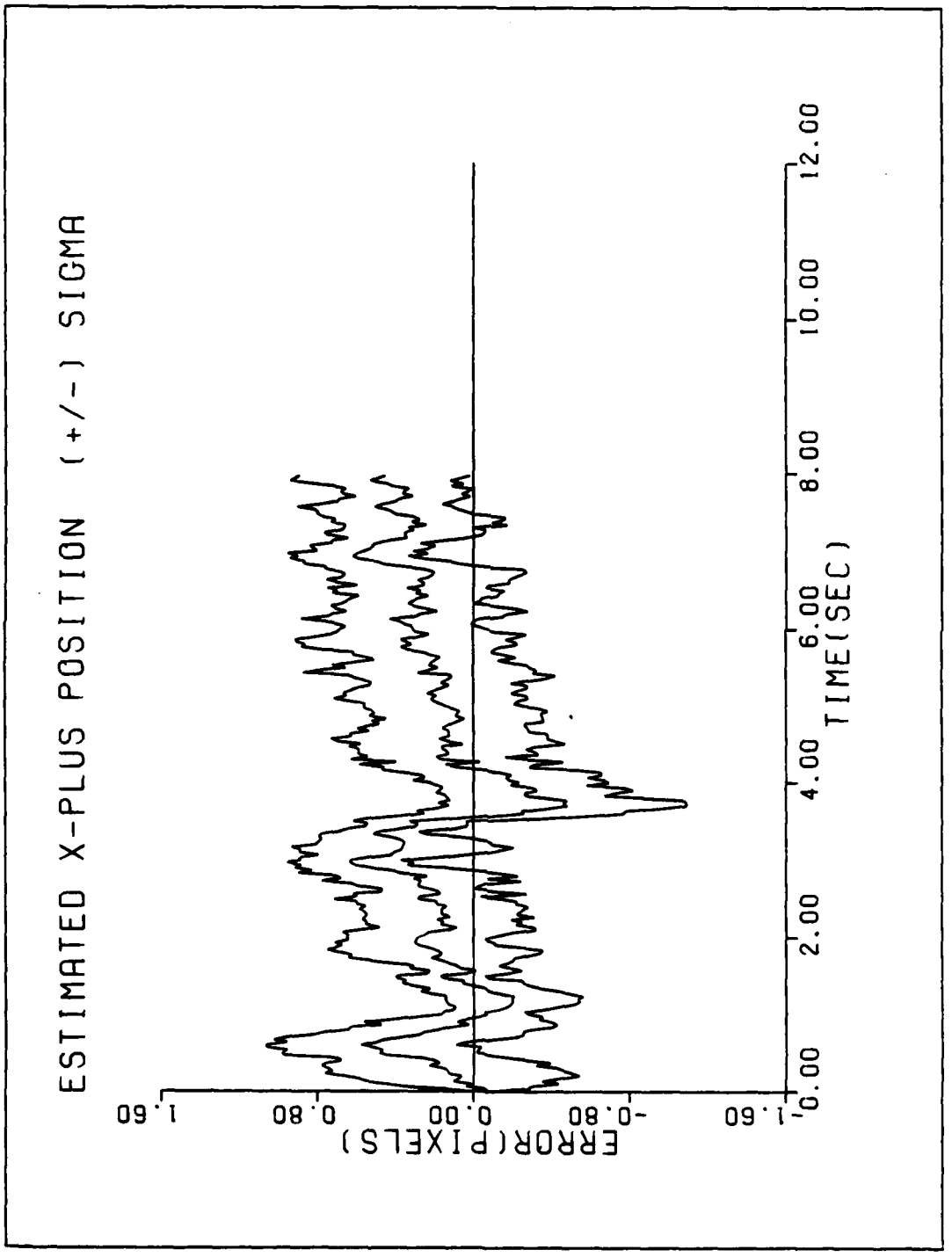


Figure D-5d. Performance Plot for T3G10MM-XT

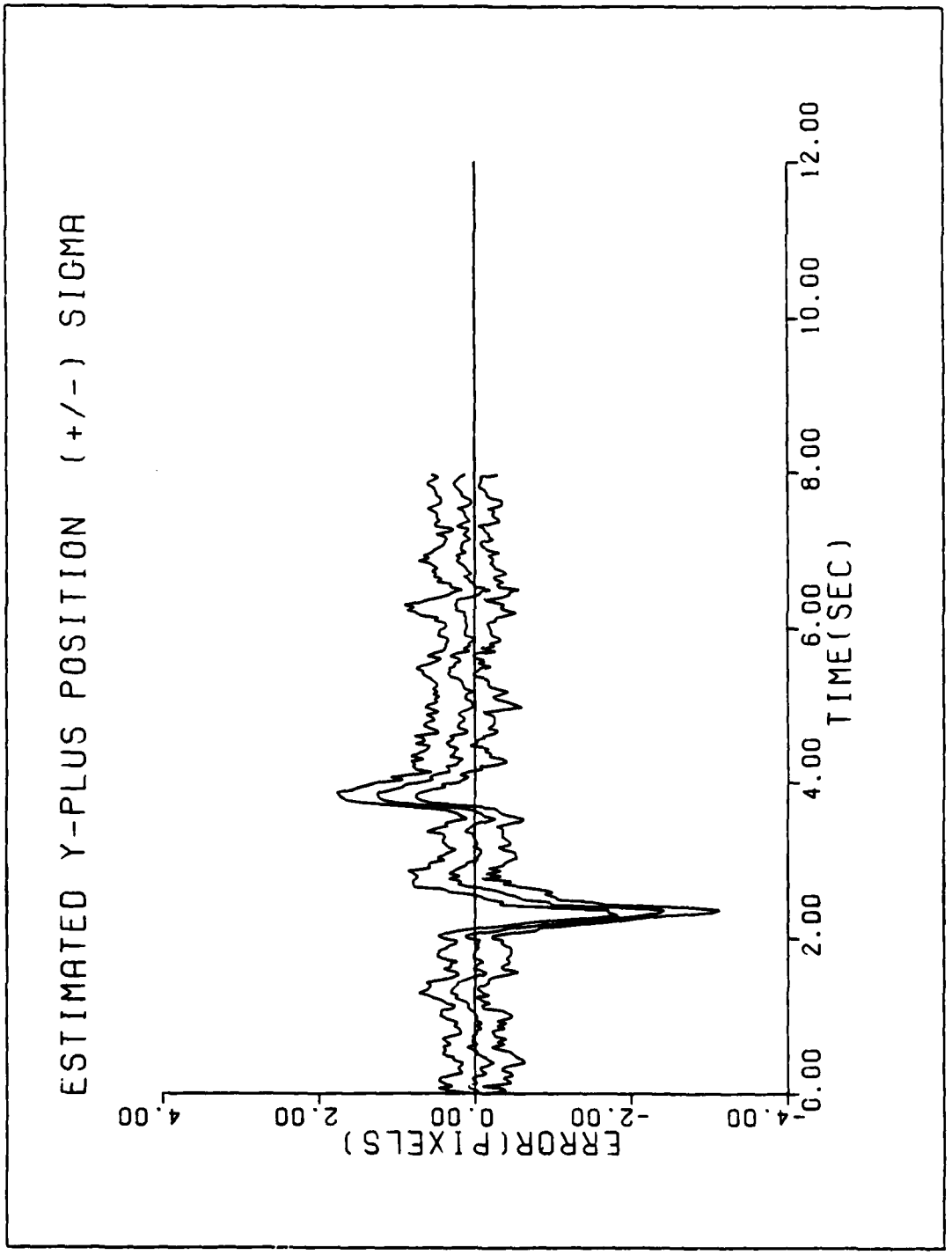


Figure D-5e. Performance Plot for T3G10MM-XT

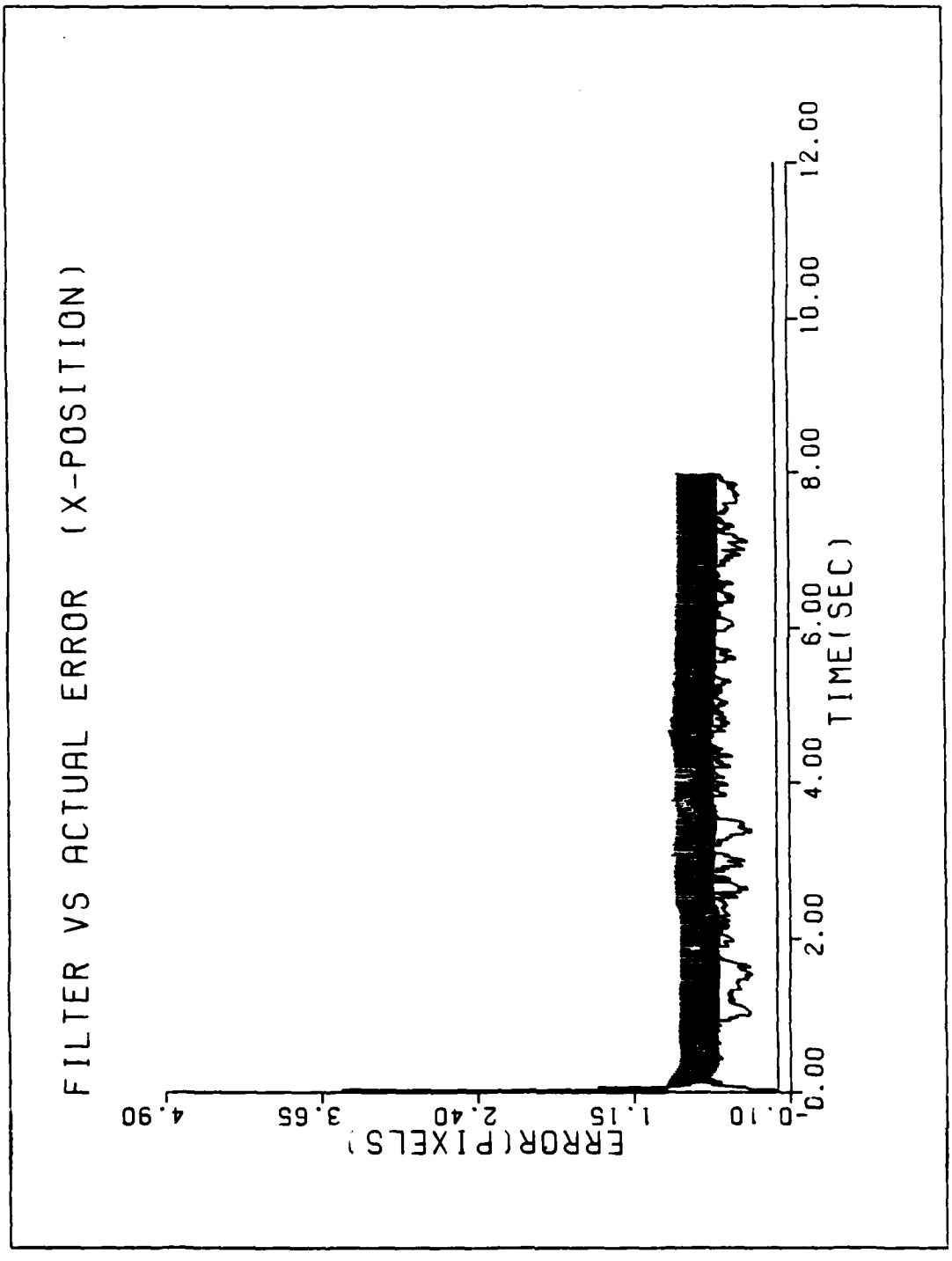


Figure D-5f. Performance Plot for T3G10MM-XT

APPENDIX E

Performance Plots for the Baseline *MMAF - Operated*

at 100 Hz Sample Rate

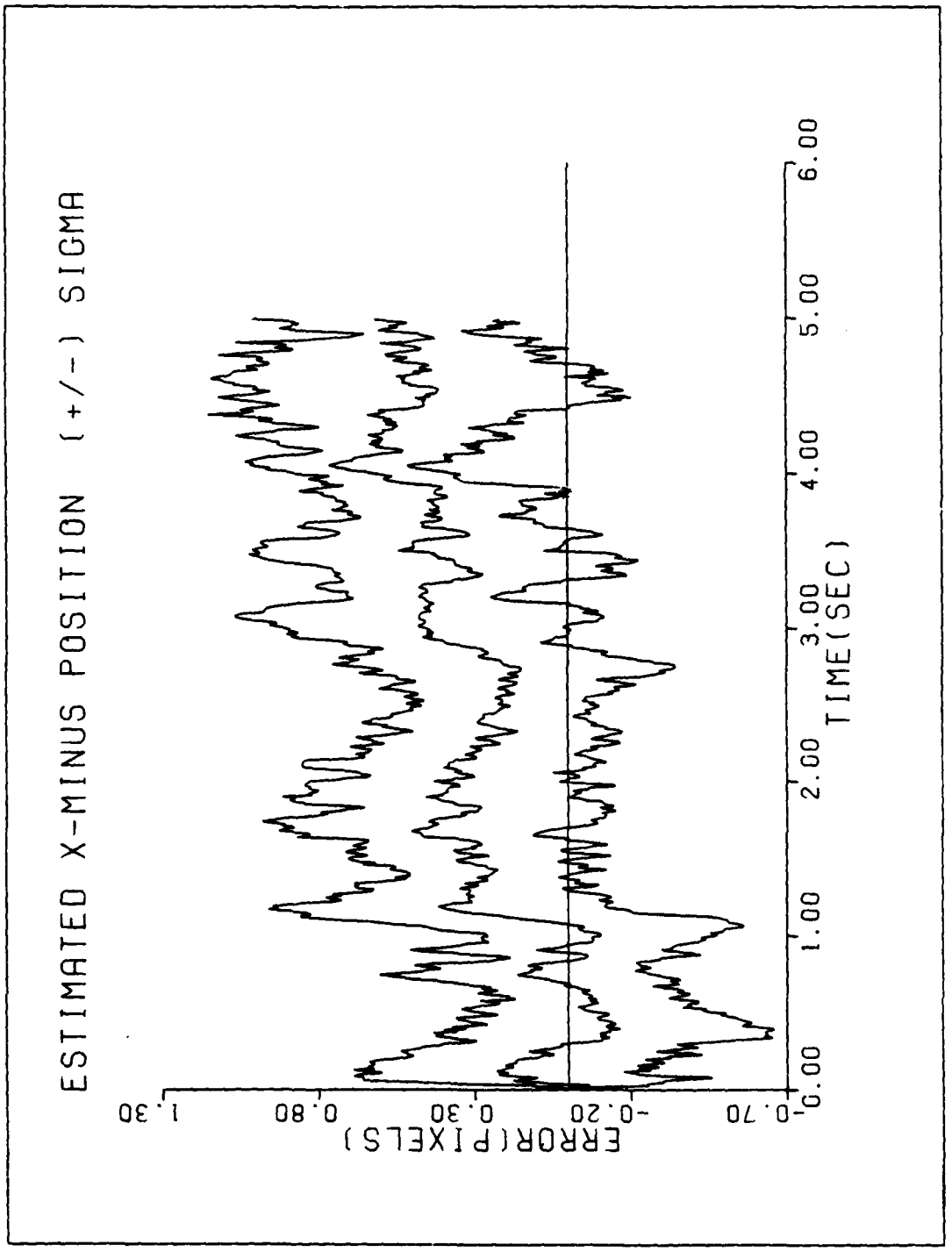
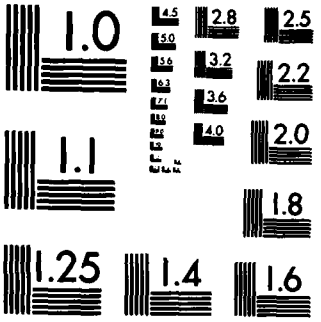


Figure E-1a. Performance Plot for TIMM-100HZ

AD-A155 466 BAYESIAN VS MAP MULTIPLE MODEL ADAPTIVE ESTIMATION FOR FIELD OF VIEW EXPA.. (U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. P A LOVING

UNCLASSIFIED MAR 85 AFIT/GE/EE/85M-1 F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

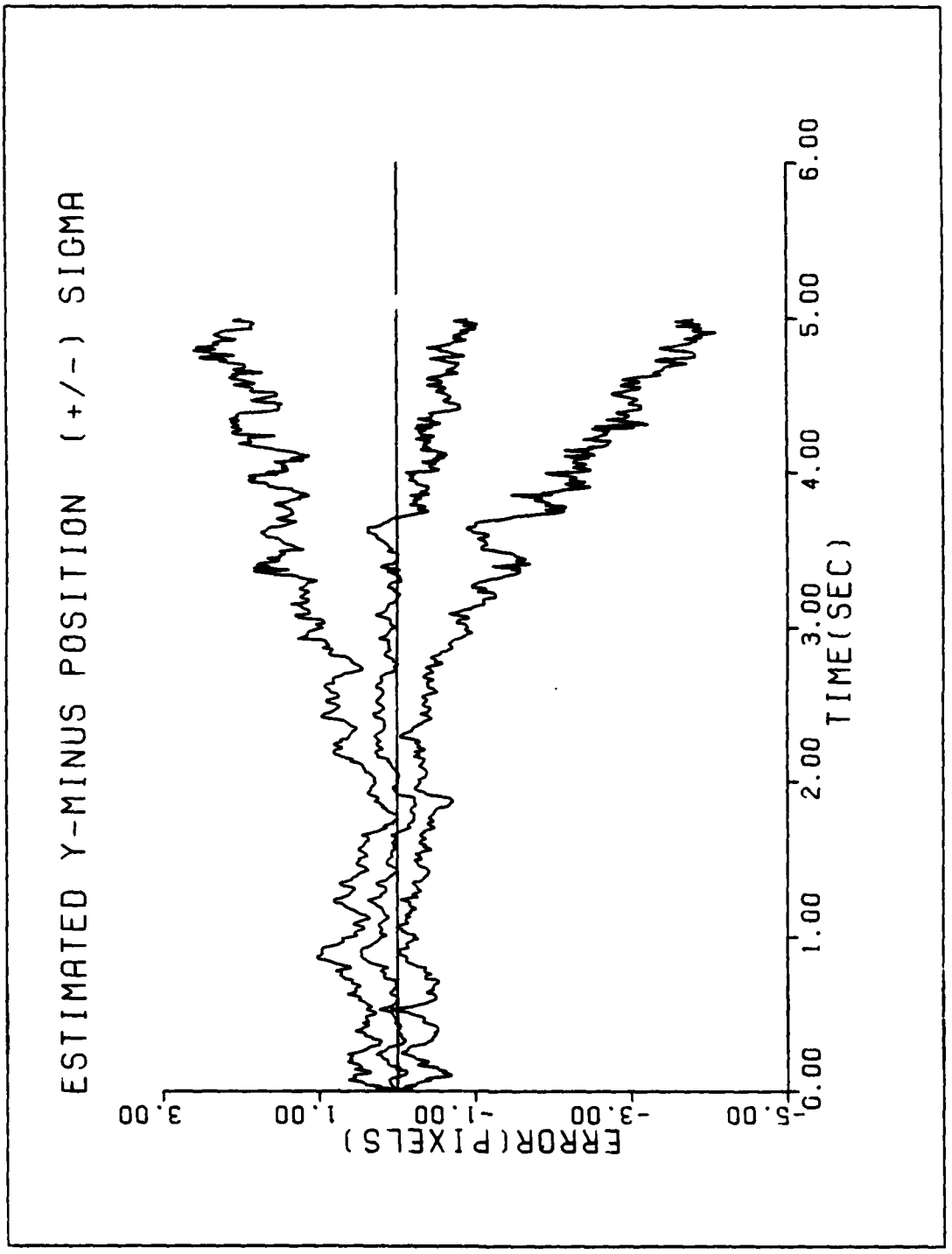


Figure E-1b. Performance Plot for TIMM-100HZ

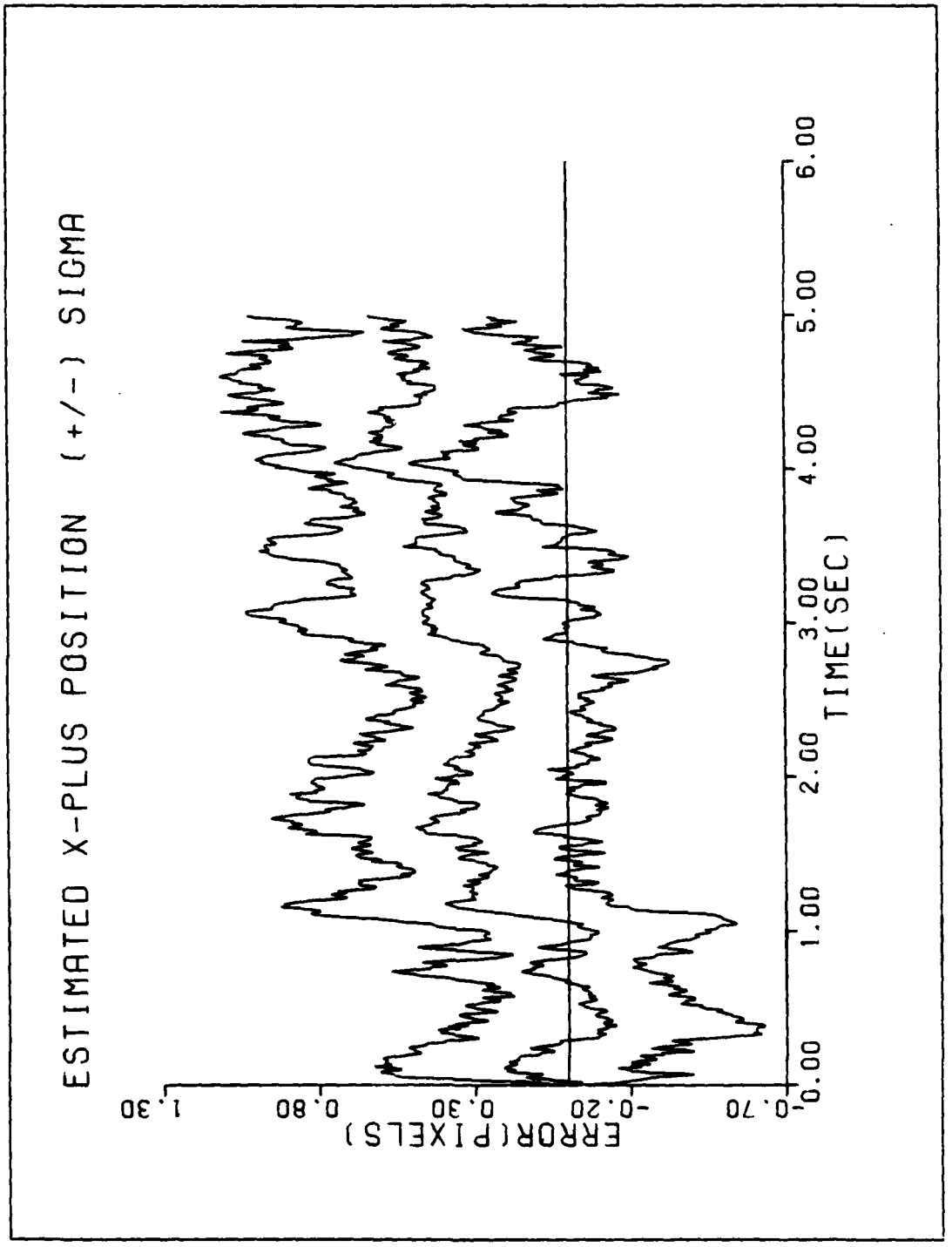


Figure E-1c. Performance Plot for TIMM-100HZ

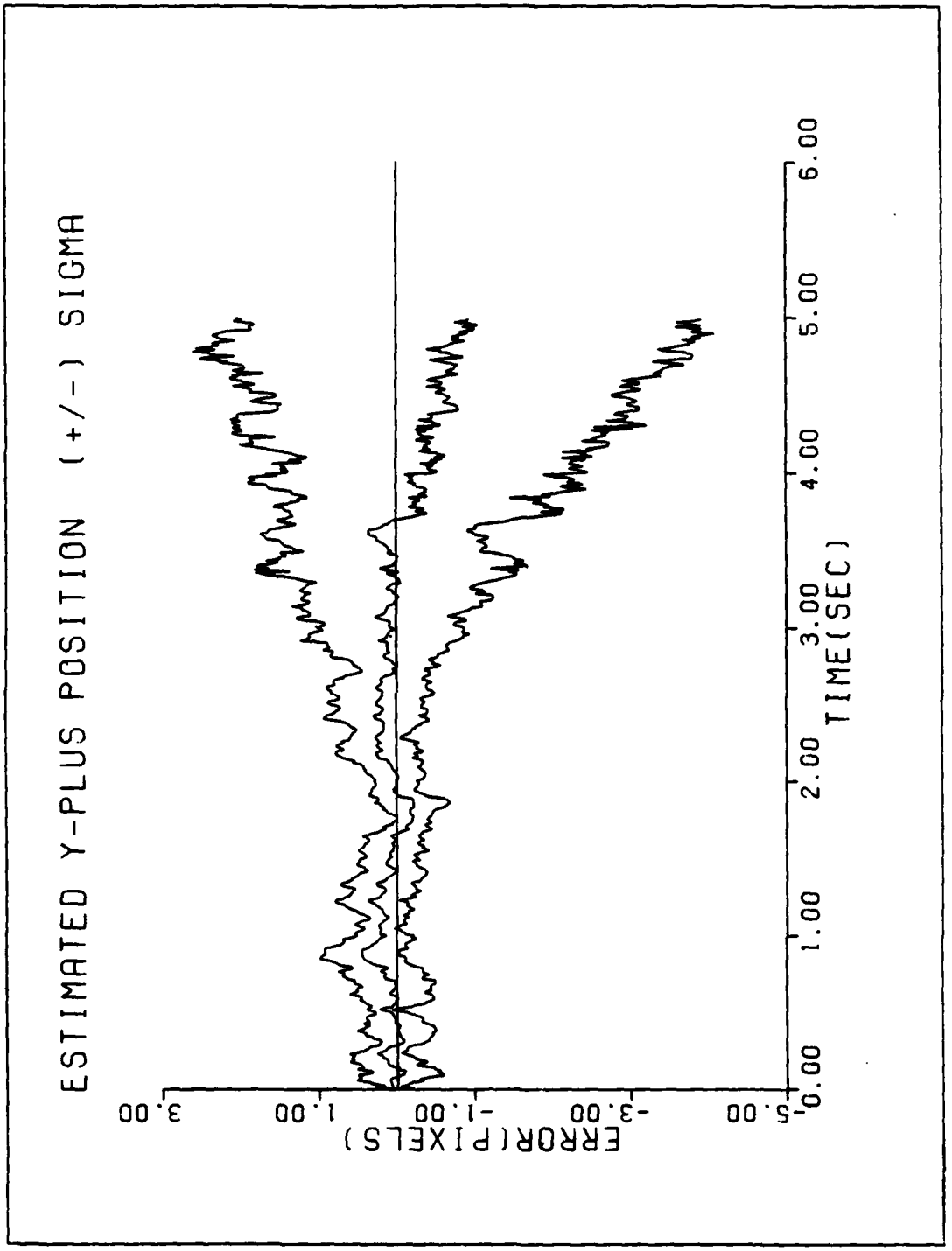


Figure E-1d. Performance Plot for TIMM-100HZ

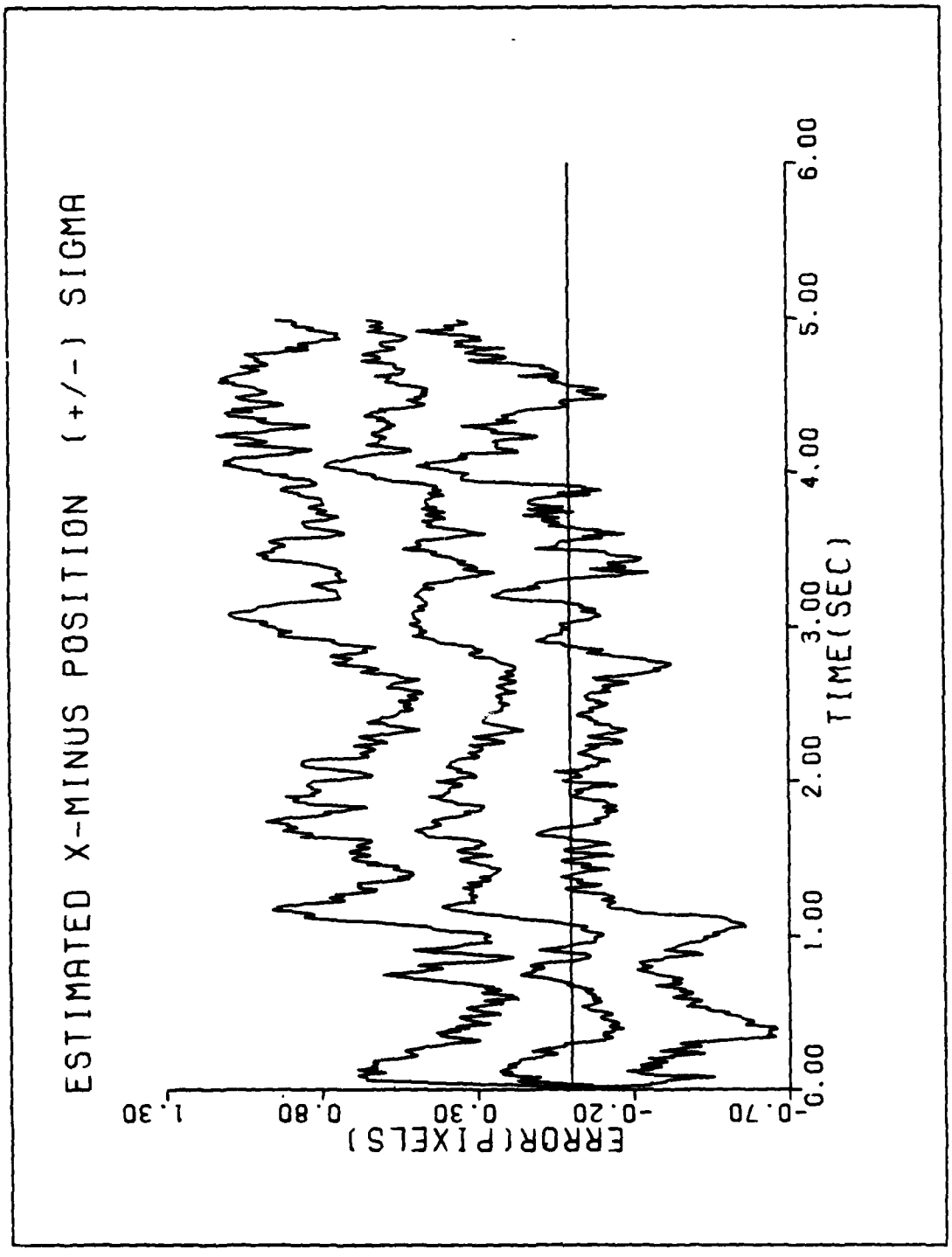


Figure E-2a. Performance Plot for T2G2MM-100HZ

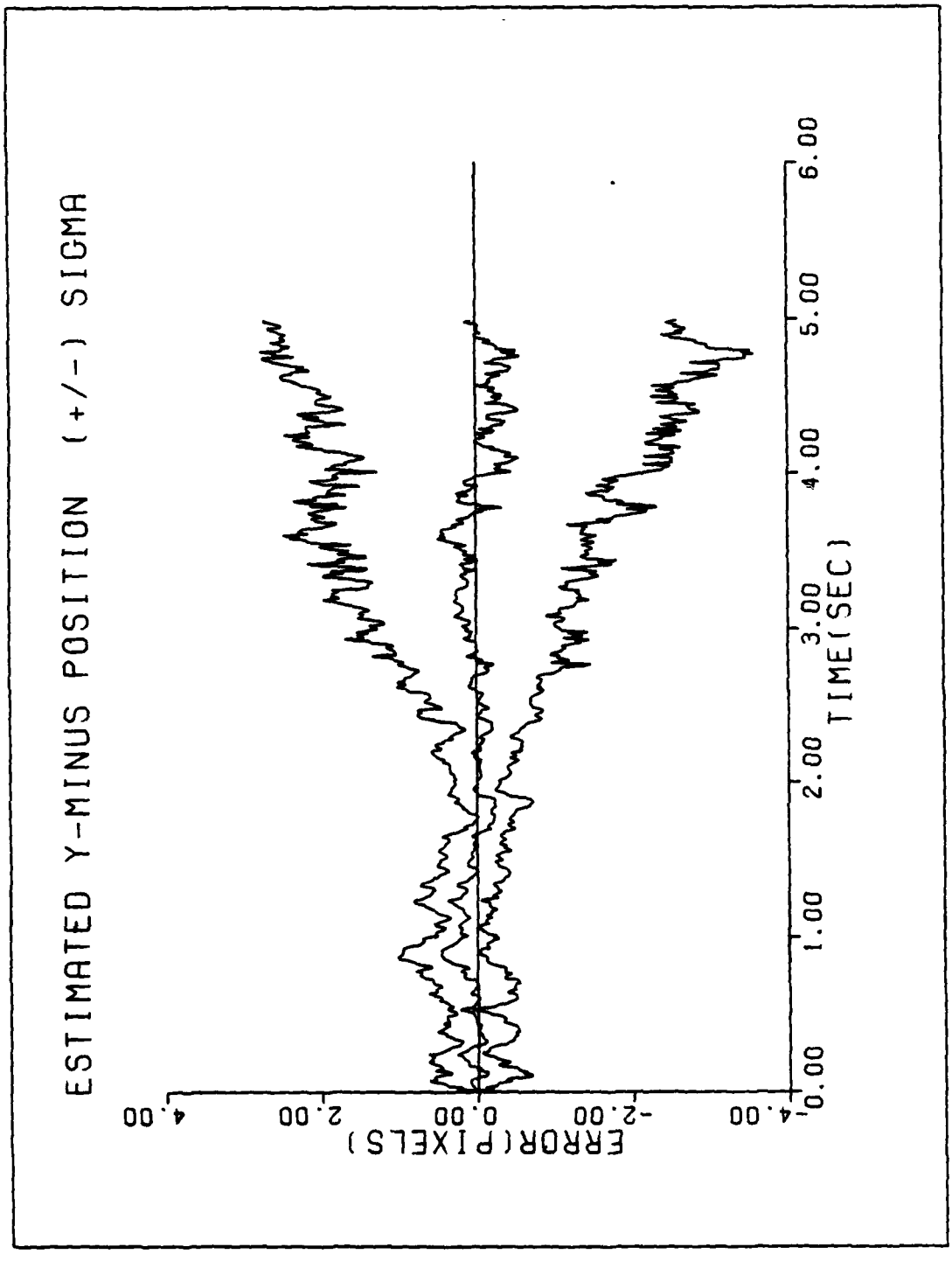


Figure E-2b. Performance Plot for T2G2MM-100HZ

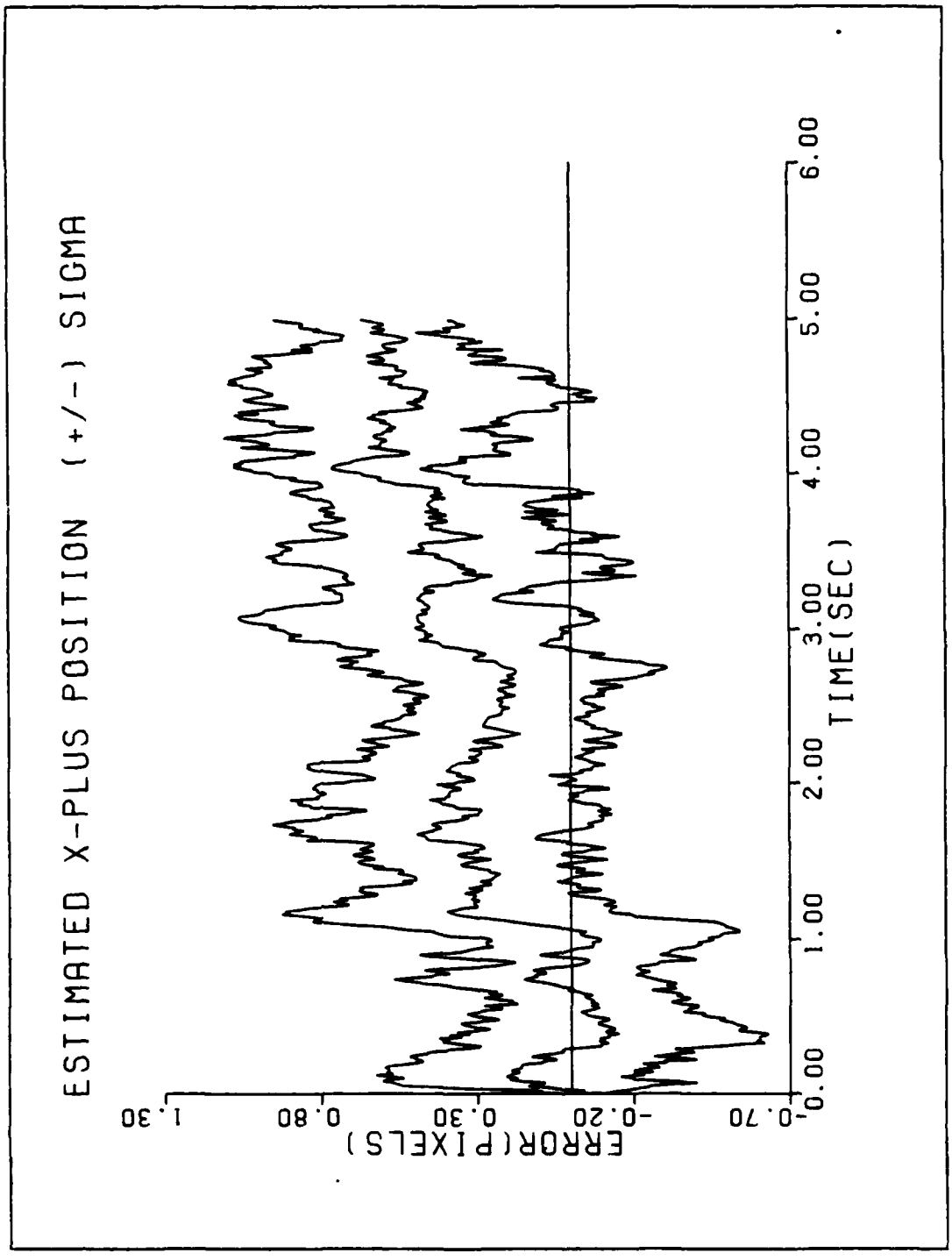


Figure E-2c. Performance Plot for T2G2MM-100HZ

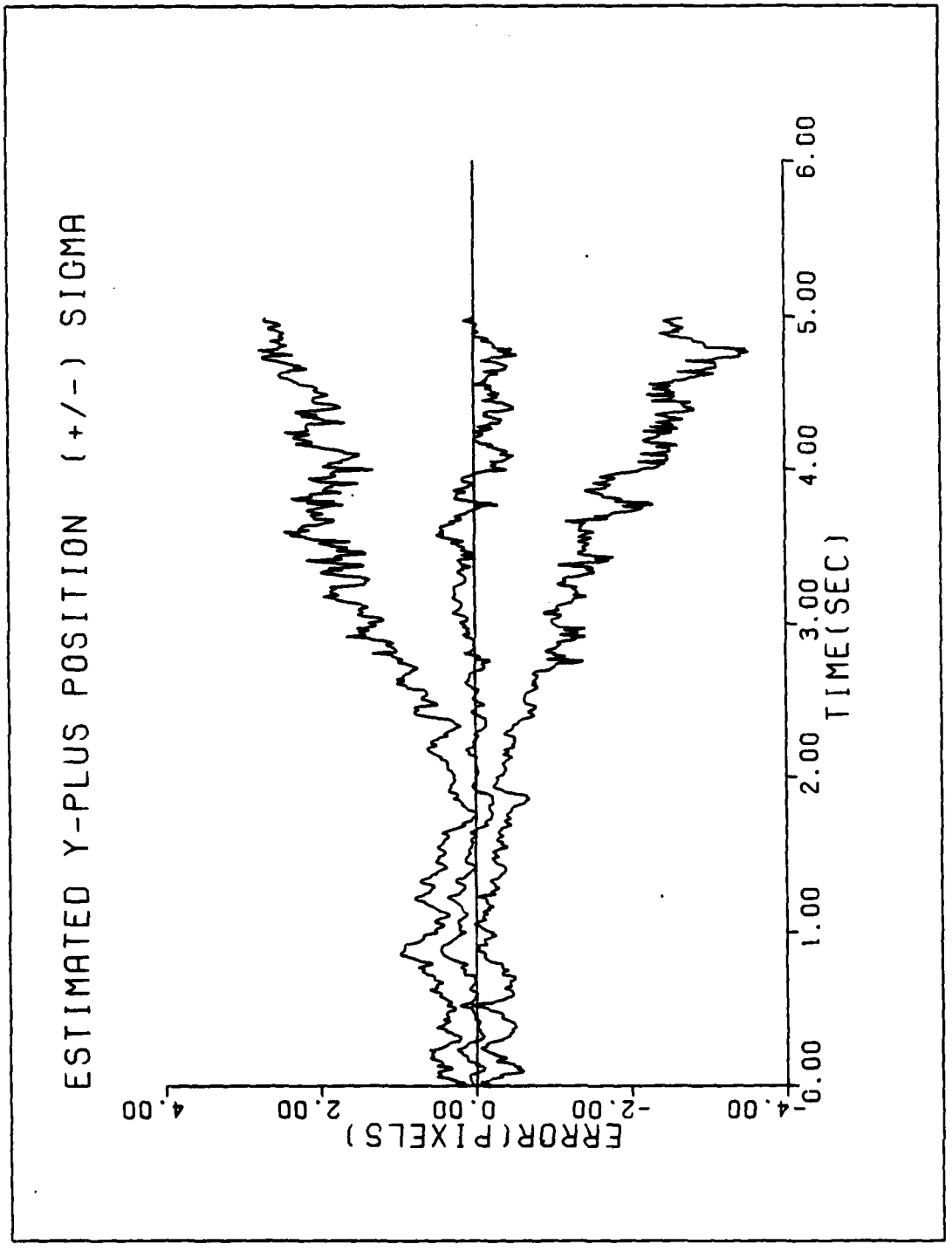


Figure E-2d. Performance Plot for T2G2MM-100HZ

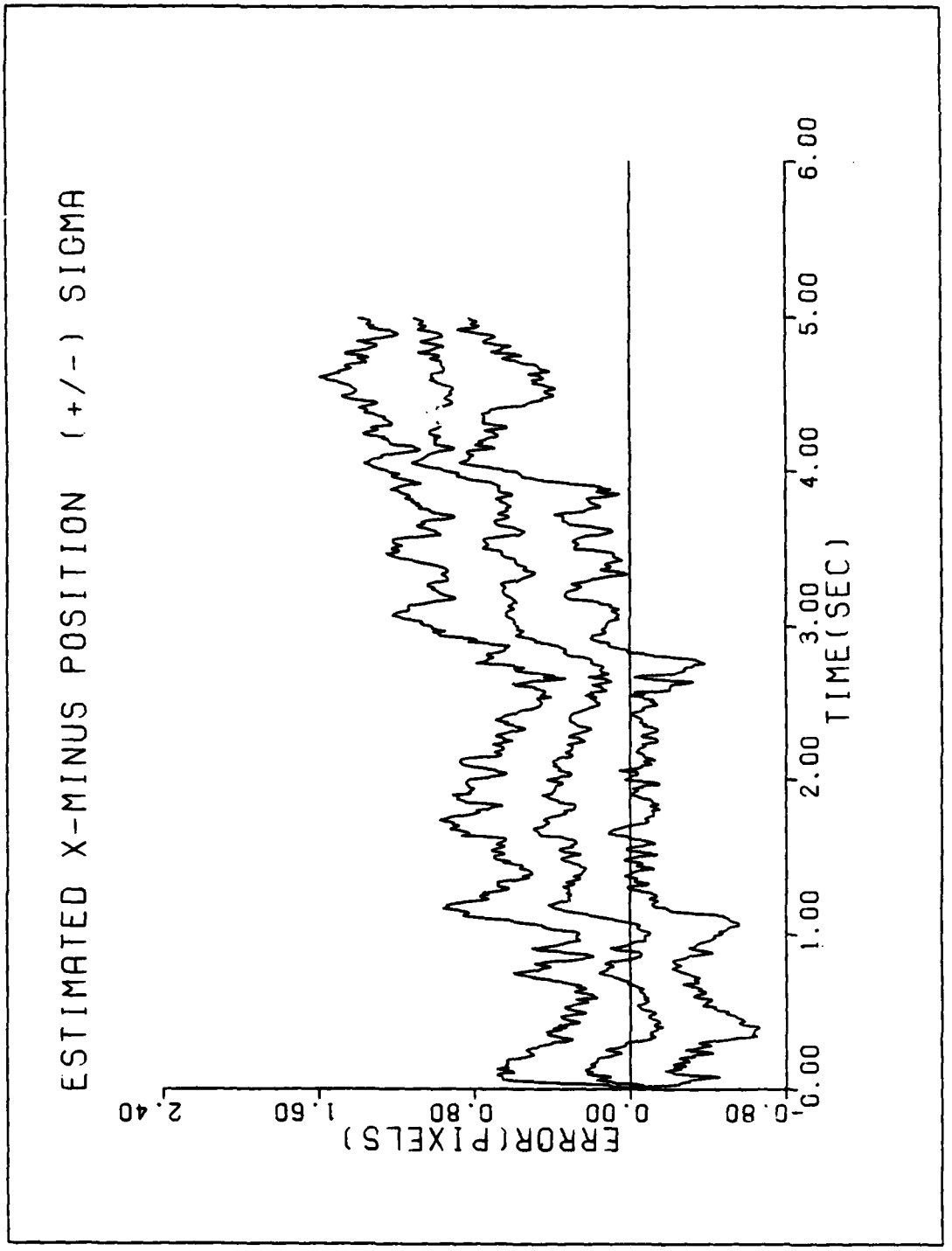


Figure E-3a. Performance Plot for T2G10MM-100HZ

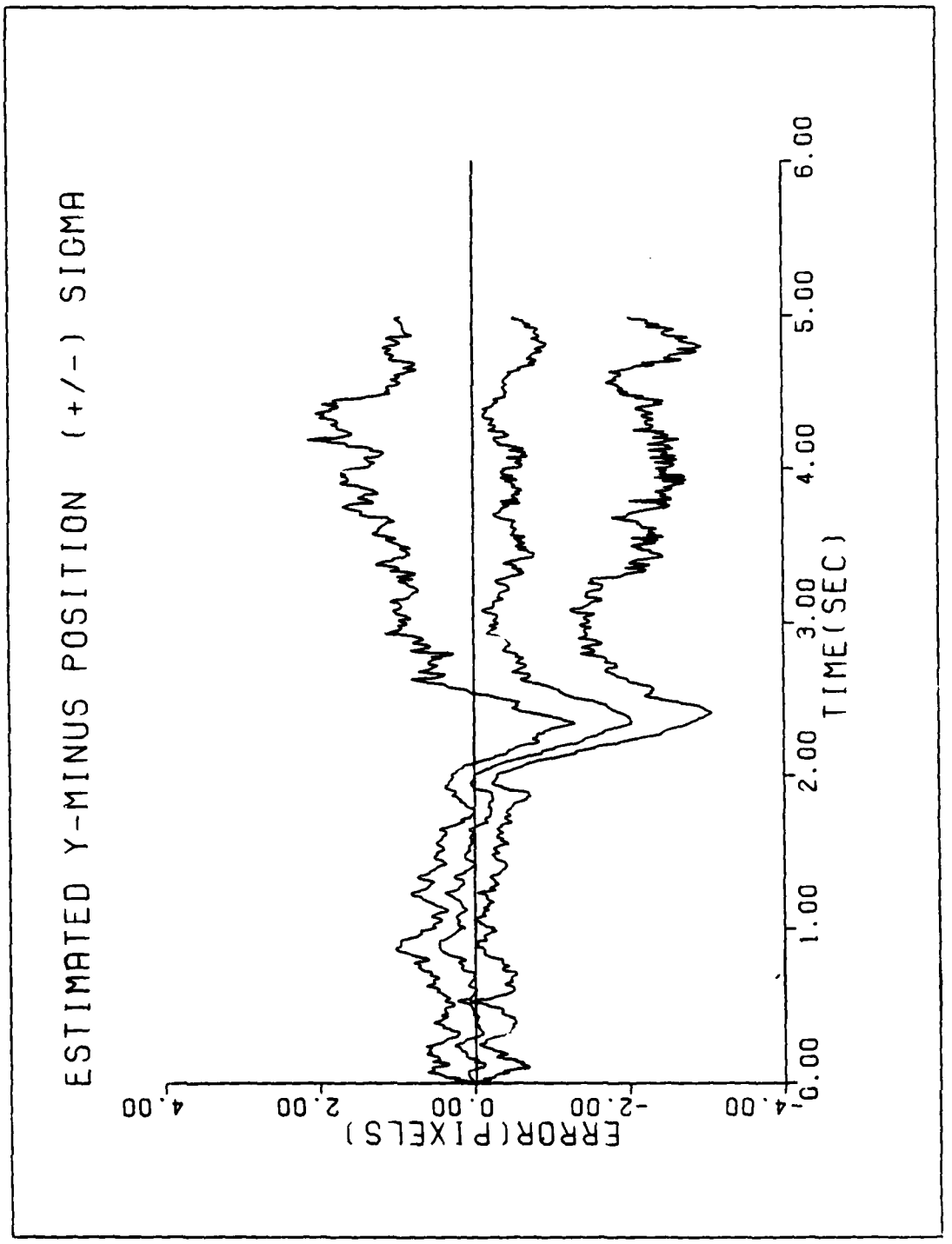


Figure E-3b. Performance Plot for T2G10MM-100HZ

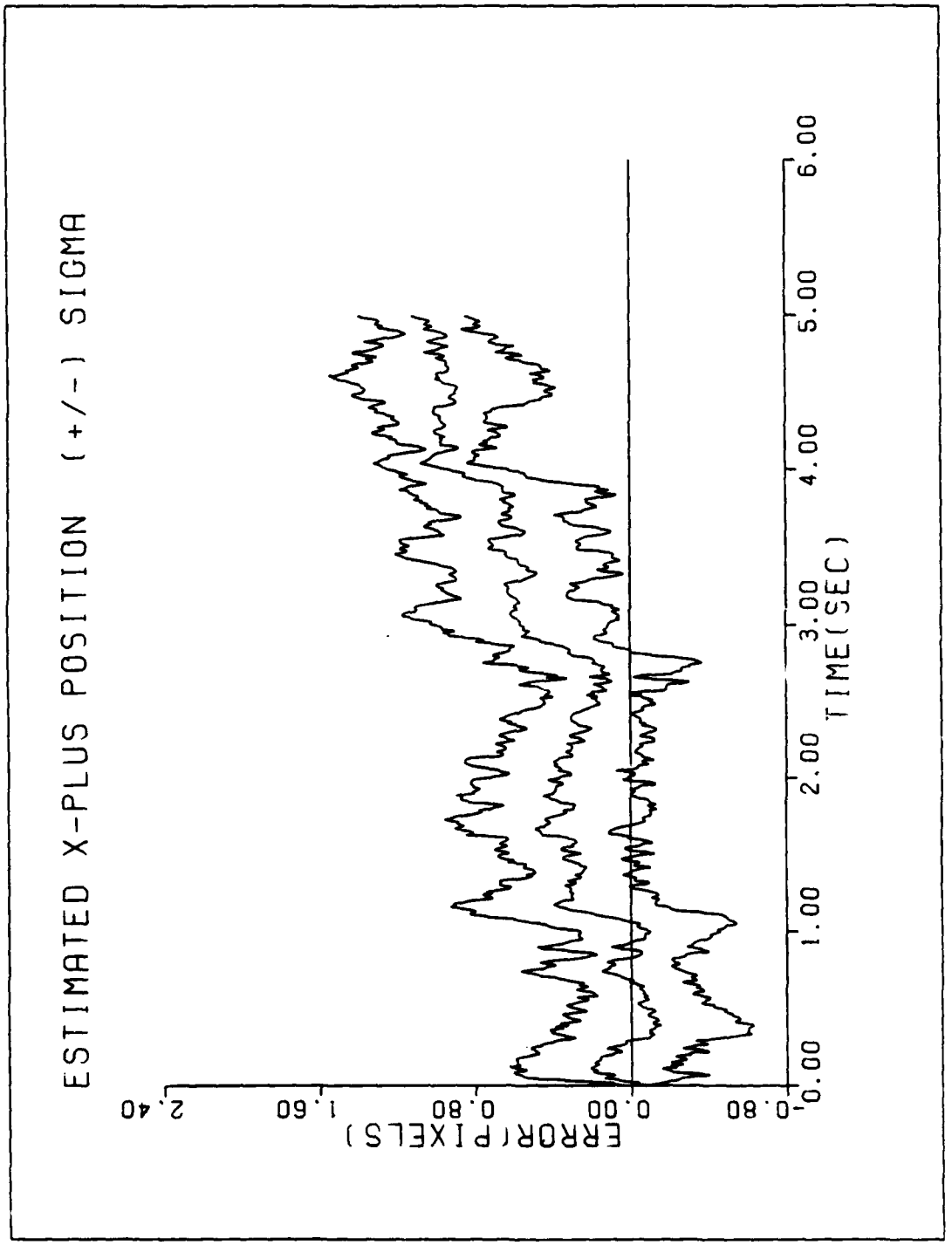


Figure E-3c. Performance Plot for T2G10MM-100HZ

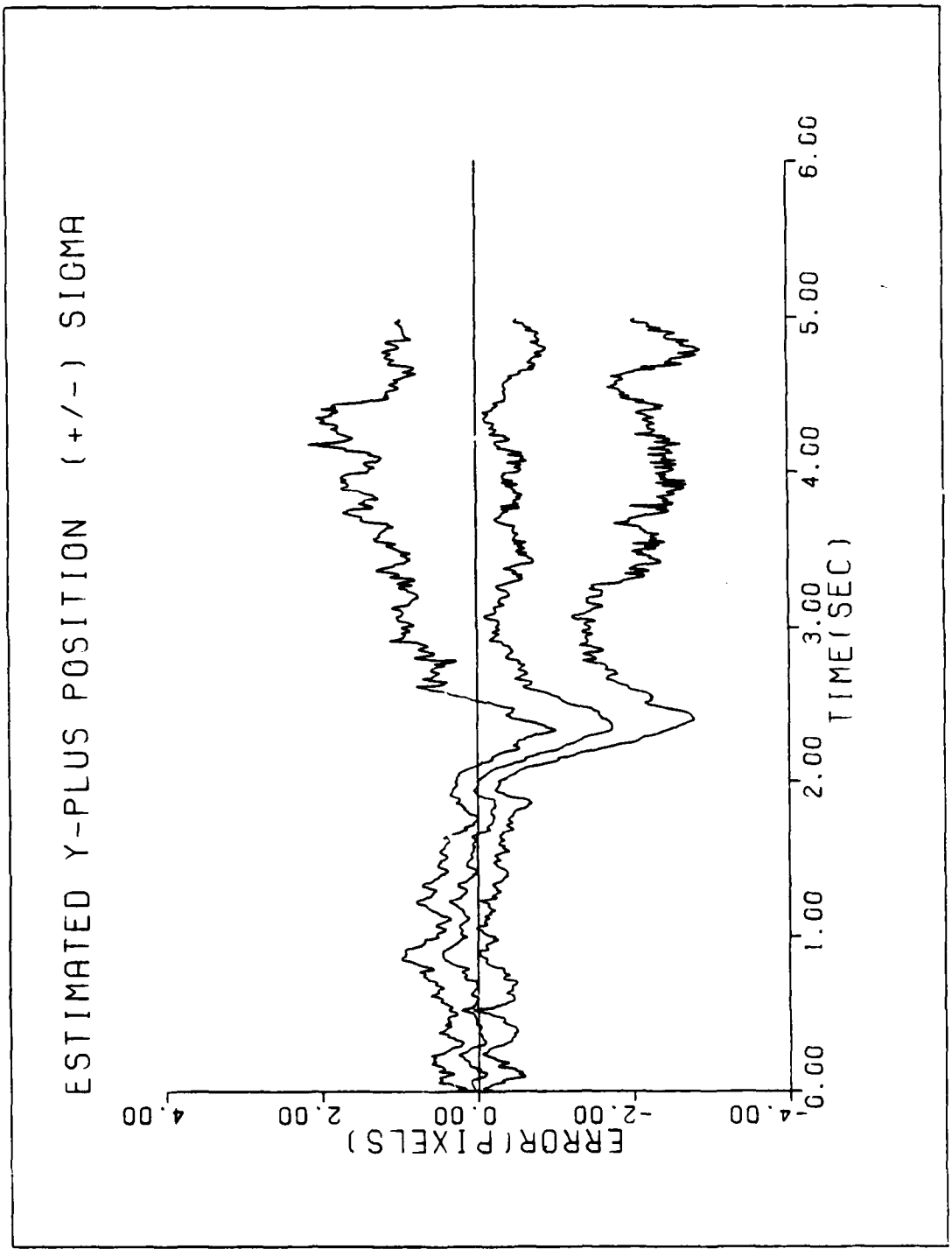


Figure E-3d. Performance Plot for T2G10MM-100HZ

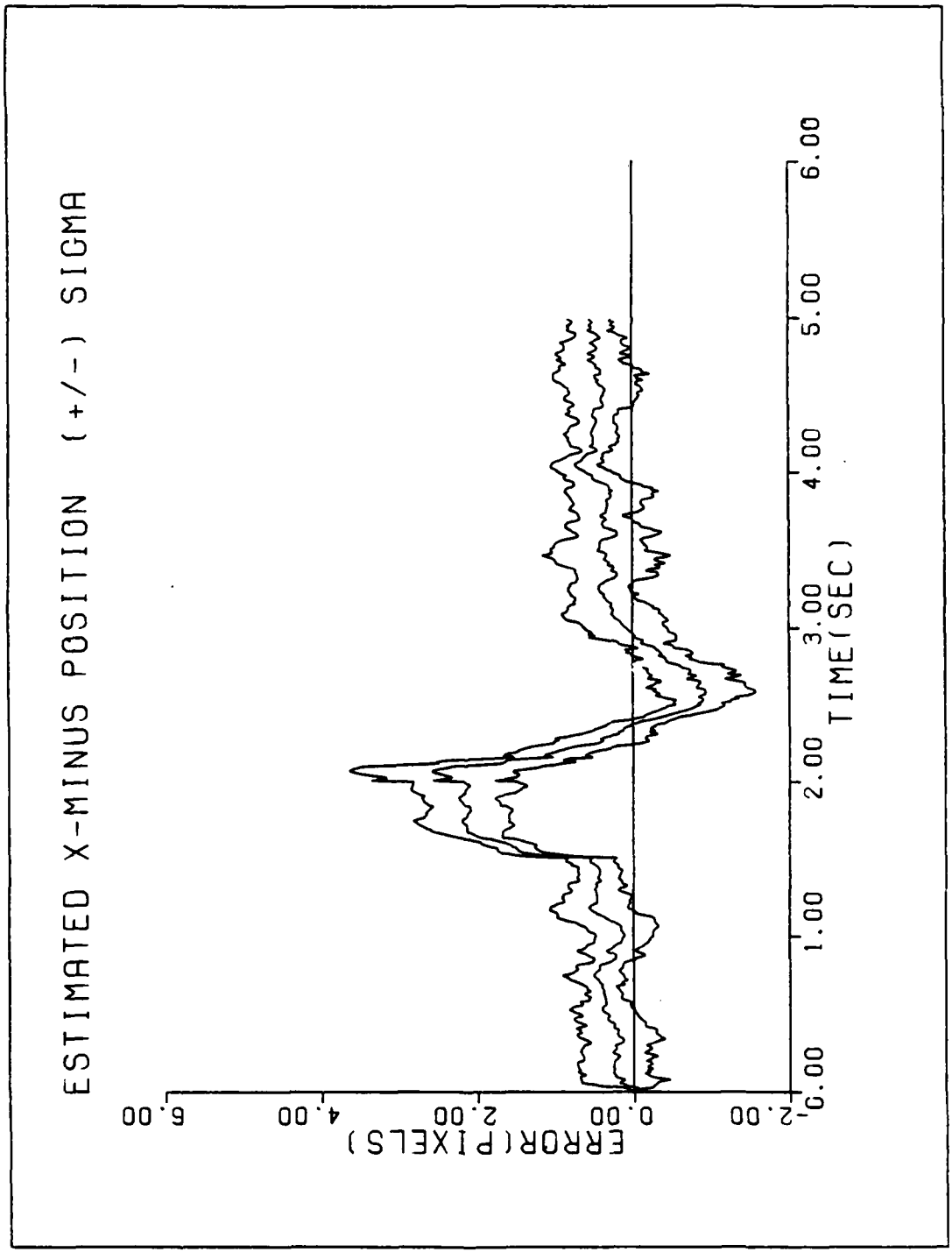


Figure E-4a. Performance Plot for T3G2MM-100HZ

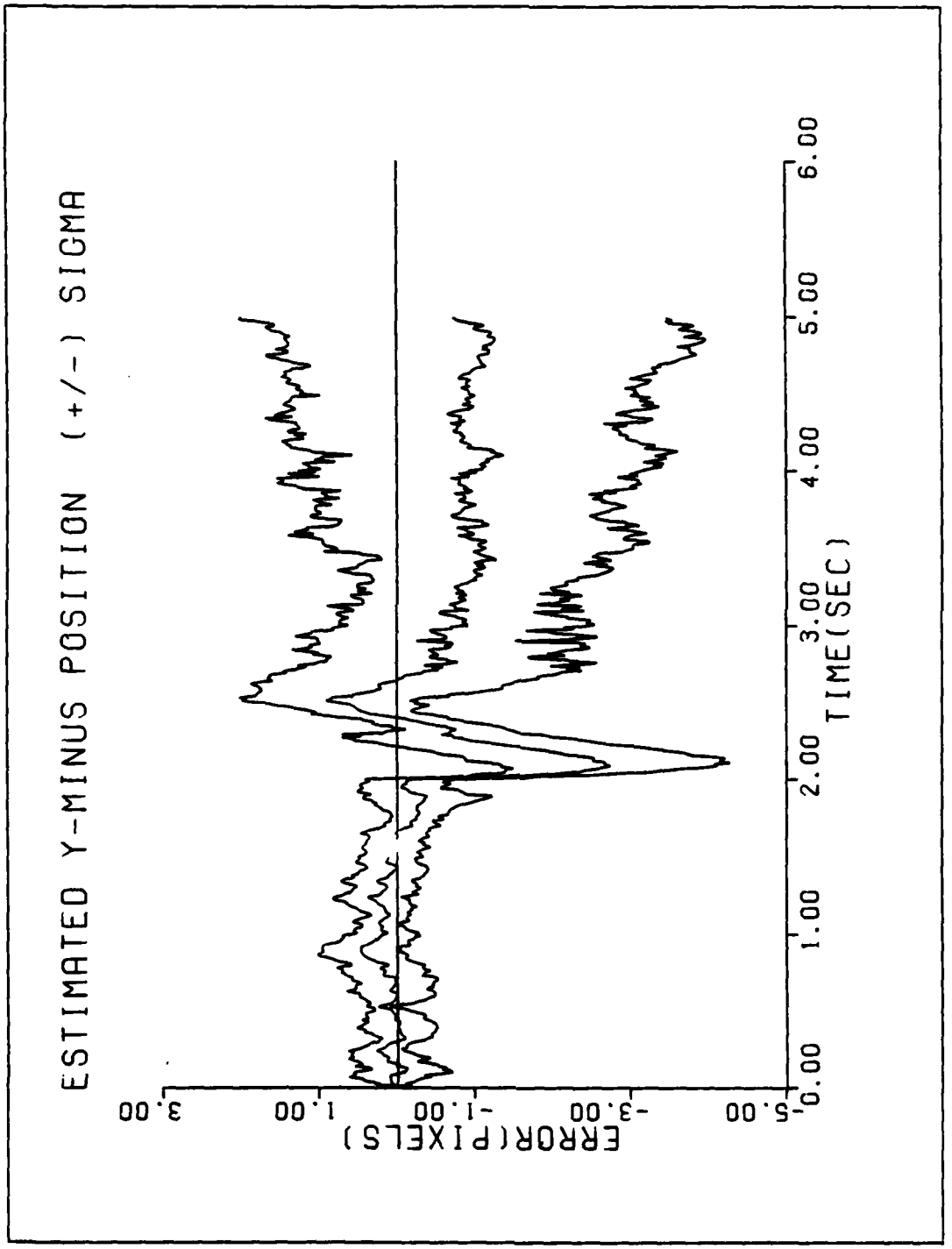


Figure E-4b. Performance Plot for T3G2MM-100HZ

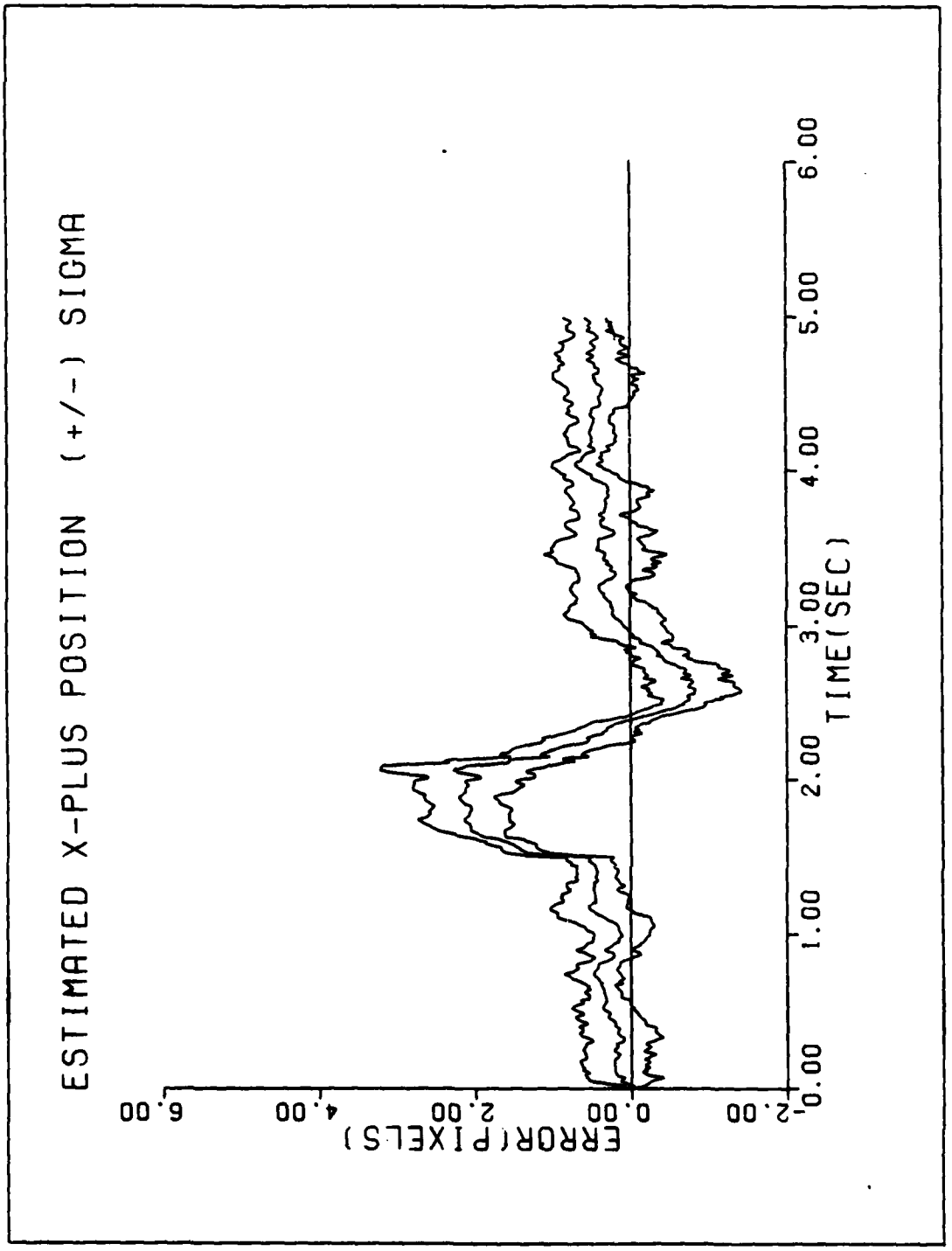


Figure E-4c. Performance Plot for T3G2MM-100HZ

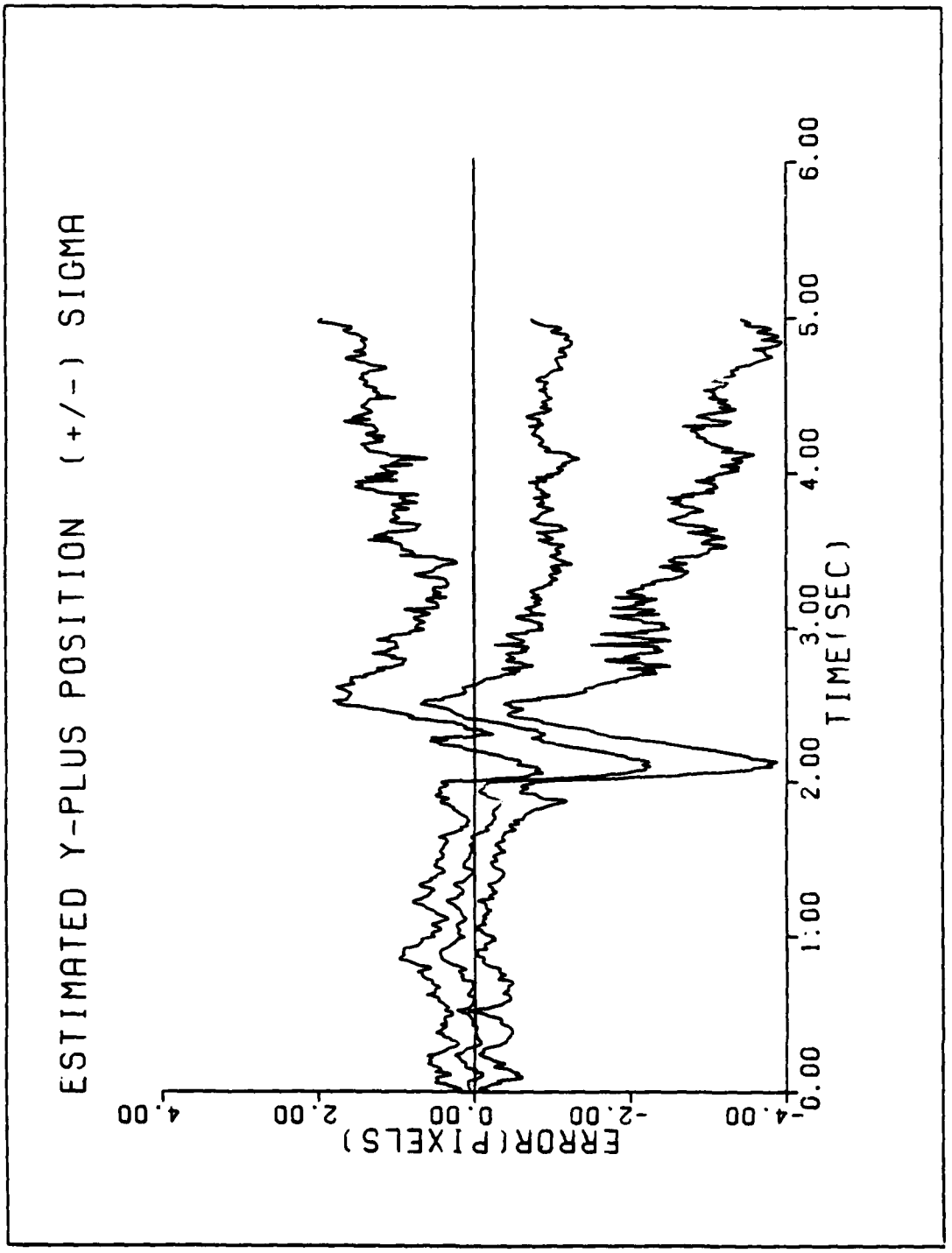


Figure E-4d. Performance Plot for T3G2MM-100HZ

As noted in Section 6.2.2, the original code for the 100 Hz sample rate resulted in loss of lock. Subsequent investigation revealed that the reacquisition revealed that the reacquisition subroutine had not been implemented correctly. The original code reset the covariance matrix to the initial values, which are much higher than steady state values and have no cross-terms. The higher than appropriate values caused a very low emphasis on incoming data at a point when the measurements should be weighted heavily. When the code is correctly implemented, the MMAF does not lose lock, as shown in the following plots of trajectory two at 20 g's. At this point, there is no explanation for the significant bias following the maneuver recovery.

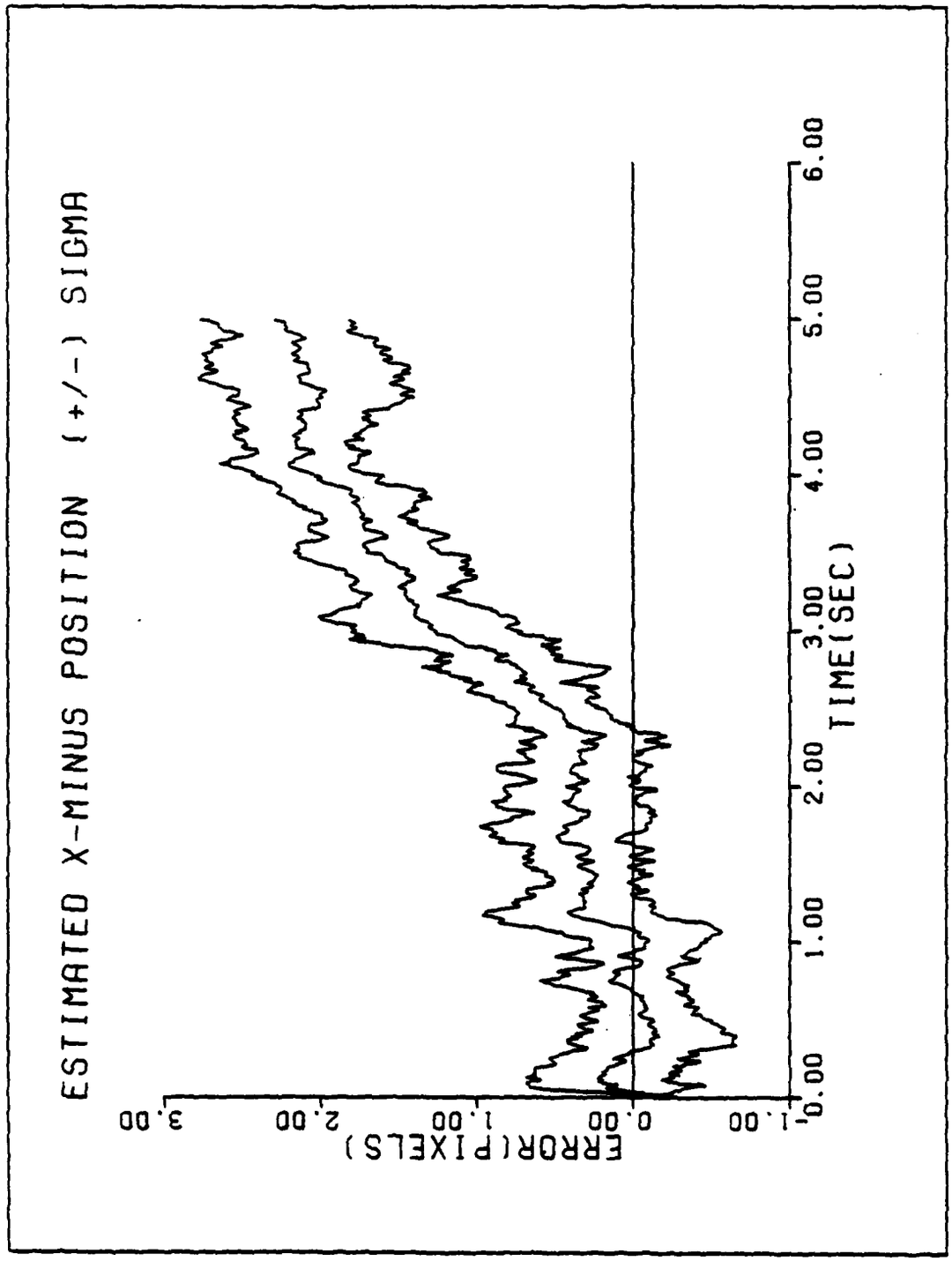


Figure E-5a. Performance Plot for T2G20MM-100HZ

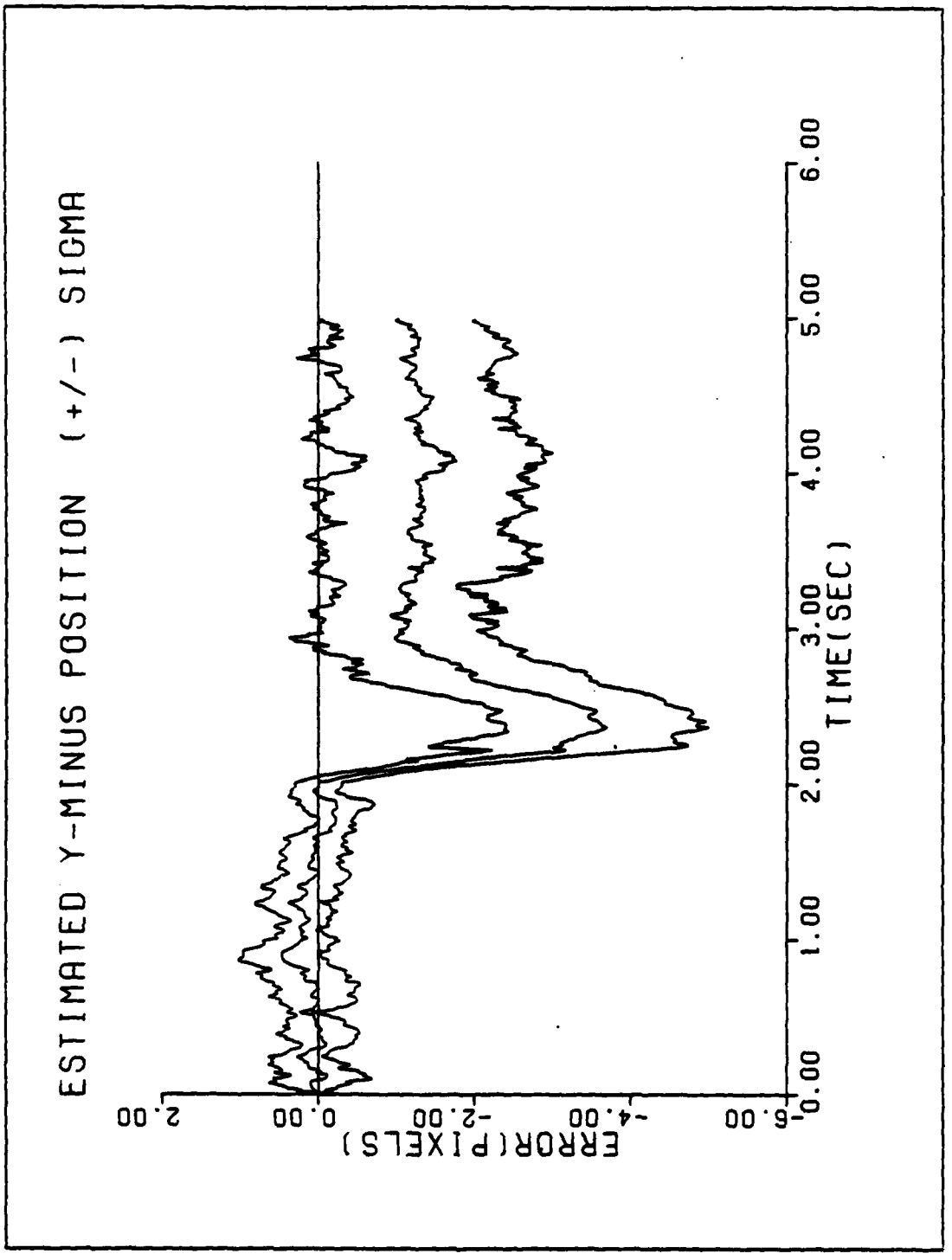


Figure E-5b. Performance Plot for T2G20MM-100HZ

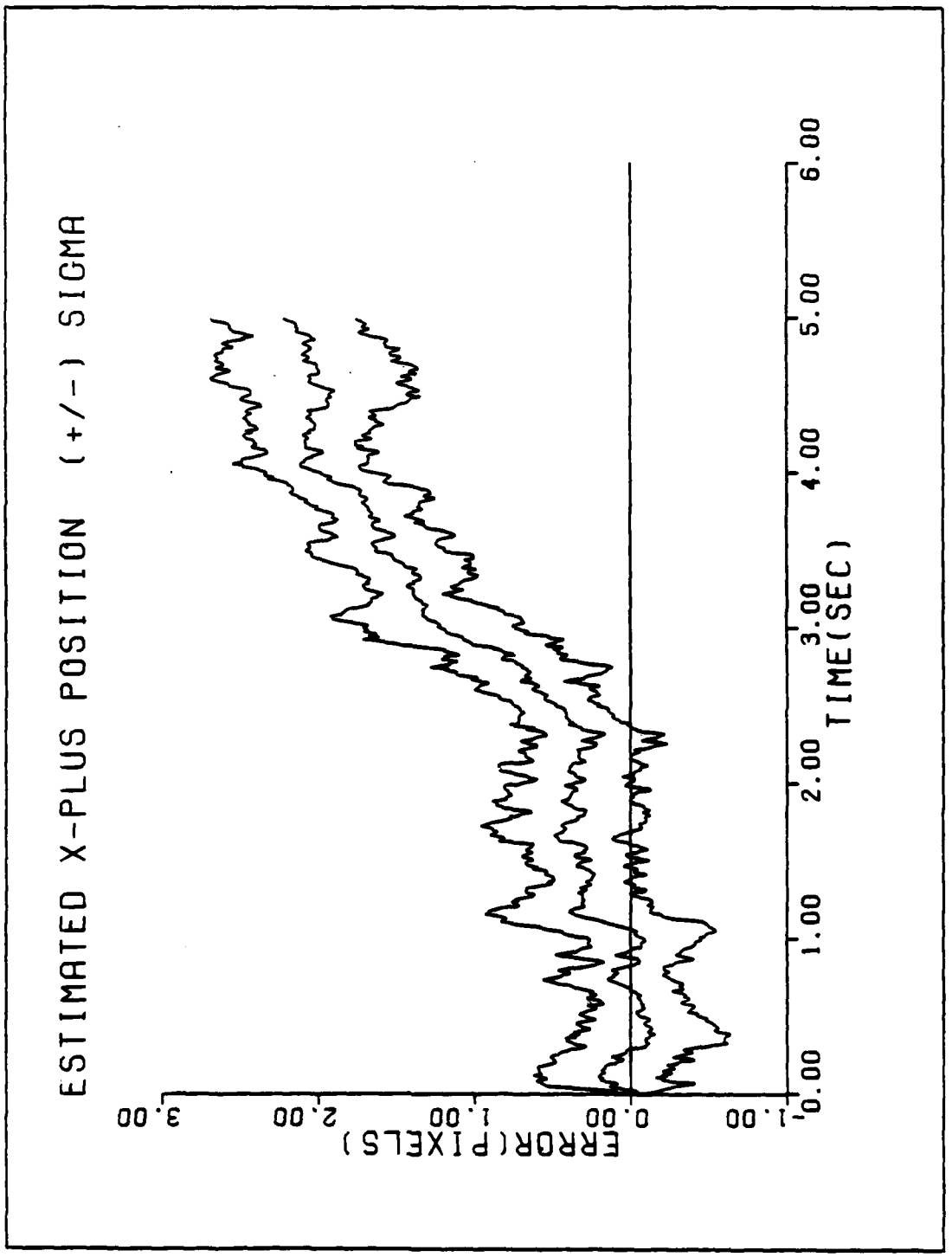


Figure E-5c. Performance Plot for T2G20MM-100HZ

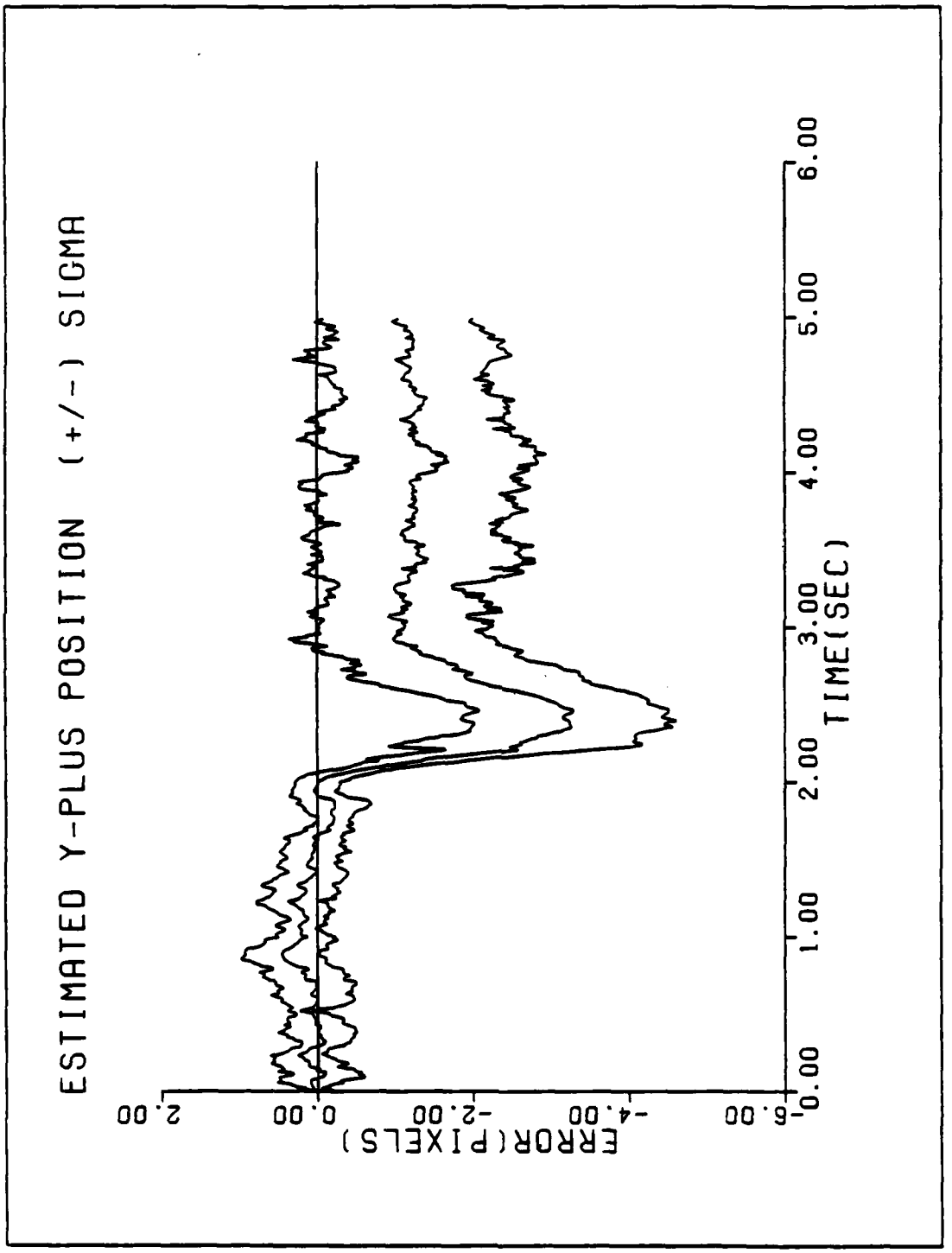


Figure E-5d. Performance Plot for T2G20MM-100HZ

Bibliography

1. ALPHATECH, Inc. Application of Multiple Model Adaptive Estimation Algorithms to Maneuver Detection and Estimation. TR-152. Prepared for U.S. Army Armament Research and Development Center (DRSMC-TSS), June 1983.

2. Flynn, Patrick M., "Alternative Dynamics Models and Multiple Model Filtering for a Short Range Tracker," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1981.

3. Harnly, Douglas A. and Robert L. Jensen, "An Adaptive Distributed-Measurement Extended Kalman Filter for a Short Range Tracker," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1979.

4. Kozemchak, Mark R., "Enhanced Image Tracking: Analysis of Two Acceleration Models in Tracking Multiple Hot-Spot Images," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1982.

5. Maybeck, Peter S. Stochastic Models, Estimation, and Control Volume I. New York: Academic Press Incorporated, 1979.

6. Maybeck, Peter S. Stochastic Models, Estimation, and Control Volume II. New York: Academic Press Incorporated, 1982.

7. Mercier, Daniel E., "An Extended Kalman Filter for Use in a Shared Aperture Medium Range Tracker," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1978.

8. Millner, Paul P., "Enhanced Tracking of Airborne Targets Using a Correlator/Kalman Filter," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1982.

9. Oppenheim, Alan V. Applications of Digital Signal Processing. New Jersey: Prentice-Hall Incorporated, 1973.

10. Roemer, W.A., "An Optically Implemented Kalman Filter Algorithm," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1983.

11. Rogers, Steven K., "Enhanced Tracking of Airborne Targets Using Forward Looking Infrared Measurements," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1981.

12. Singletery, James Jr., "Adaptive Laser Pointing and Tracking Problem," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1980.

13. Suizu, Robert I., "Enhanced Tracking of Airborne Targets Using Multiple Model Filtering Techniques for Adaptive Field of View Expansion," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1983.

Vita

Captain Phyllis A. Loving was born on 11 October 1958 in Austin, Texas. She graduated from high school in Colorado Springs, Colorado, in 1976. In May 1980, she graduated from the United States Air Force Academy with a Bachelor Science degree majoring in Operations Research and General Engineering. From June 1980 to May 1983, she served as a systems engineer in the Space Defense Systems Program Office at Space Division, Los Angeles Air Force Station. Captain Loving entered the Air Force Institute of Technology in May 1983 to pursue a Master of Science degree in Electrical Engineering. Captain Loving is a member of Eta Kappa Nu and Tau Beta Pi.

Permanent Address:

569 36th Street
Manhattan Beach, CA 90266

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GE/EE/85M-1	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: School of Engineering, Air Force Institute of Tech	6b. OFFICE SYMBOL (If applicable): AFIT/EN
6c. ADDRESS (City, State and ZIP Code): Wright-Patterson AFB, Ohio 4543?, Air Force Institute of Technology	7a. NAME OF MONITORING ORGANIZATION
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Air Force Weapons Lab	8b. OFFICE SYMBOL (If applicable): AFWL/ARAA
8c. ADDRESS (City, State and ZIP Code): Kirtland AFB, NM 87117	7b. ADDRESS (City, State and ZIP Code)
11. TITLE (Include Security Classification): See Box 19	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
	10. SOURCE OF FUNDING NOS. — PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
12. PERSONAL AUTHOR(S): Phyllis A. Loving, B.S., Capt., USAF	
13a. TYPE OF REPORT: MS Thesis	13b. TIME COVERED: FROM _____ TO _____
14. DATE OF REPORT (Yr., Mo., Day): 1985, March	15. PAGE COUNT: 310
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES		
FIELD	GROUP	SUB. GR.
17	05	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Adaptive Estimation, Image Tracking, Kalman Filtering, Target Tracking, Multiple Model Adaptive Filter, Maximum a Posteriori estimator, Forward Looking Infrared(FLIR)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title:
BAYESIAN VS MAP MULTIPLE MODEL ADAPTIVE ESTIMATION FOR FIELD OF VIEW EXPANSION IN TRACKING AIRBORNE TARGETS

Thesis Advisor: Dr. Peter S. Maybeck

Approved for public release: IAW AFR 190-17.
1 May 85
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (AIC)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
Professor of Dr. Peter S. Maybeck, Electrical Engineering	513-255-3450	AFIT/ENG

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

Abstract

Previous efforts at the Air Force Institute of Technology have led to the development of a multiple model adaptive filter (MMAF) tracking algorithm which demonstrates significant improvements in performance against close-range, highly dynamic, airborne targets, over a direct correlation method currently in use. The basic elemental filter in the MMAF bank combines an enhanced correlator and a linear Kalman filter. Digital signal processing techniques are used to derive a target shape function from the forward looking infrared sensor data. This shape function is used as a template in the correlation algorithm which generates offset pseudo-measurements for the update portion of a linear Kalman filter. The multiple models are created by tuning the basic model for "best" performance against differing target maneuvering behavior and with physically different fields of view. The outputs of three independent elemental filters, each receiving data from a shared sensor are used to generate a single adaptive estimate of the state via a probabilistic weighted average (Bayesian form) or by selection of the one elemental filter associated with the highest probability (MAP form). The adaptive state estimate can produce target position predictions to be used in generating feedback control for maintaining the target in the center of the field of view.

There are two main results from this effort. The addition of a third elemental filter to the baseline MMAF improves tracking performance over the two-element MMAF. Specifically, the peak error following a maneuver is significantly reduced. However, the MAP estimation approach does not differ significantly from the Bayesian approach.

END

FILMED

7-85

DTIC